
INPUT
Publicación práctica para usuarios de MSX
Revista mensual 1986
Año 1 - Número 2 Precio 350 Ptas.
MSX
Un poster de regalo
Impresoras plotter
El ratón y el queso
Exígele a tu Basic

AHORA TE PUEDES PROGRAMAR UN VIAJE ALUCINANTE AL EPCOT CENTER CON ALEA. ES LOGICO.

Te presentamos, en estreno mundial, todo un reto a tu inteligencia: la colección de programas Logicolor.

Con los tres juegos de la colección Logicolor tu mente desafiará a la fría lógica del ordenador. ¡Atrévete con ellos!

AUTOS LOCOS: Construye tu propia escudería y apuesta por tu bólido favorito. Un primer contacto con el uso de los símbolos. Para chicos entre 10 y 12 años. Incluye también un super-master mind contra el ordenador.

MANZANAS Y GUSANOS: Utilizando fórmulas puedes recoger las manzanas y dejar fuera los gusanos; proteger las ánforas de los golpes del martillo; defender los globos aerostáticos de las flechas enemigas; o evitar que los cañones destruyan las torres de tu fortaleza. Tu inteligencia lógica es la única arma que necesitas. Para chicos entre 12 y 14 años, y para quienes desean mantener su mente en forma.

REHENES: Tendrás que desarrollar una estrategia lógica si quieres eliminar a los conspiradores y salvar la corona. ¿Te gustaría descubrir la fórmula que abre el cofre de los diamantes? Intenta descubrir un procedimiento lógico para rescatar a los rehenes. Para chicos entre 14 y 16 años, y para quienes se las dan de genios.

Además, la compra de cada programa de la colección Logicolor te da derecho a participar en el fabuloso concurso EPCOT, y si consigues vencer al ordenador, tus posibilidades de conseguir un magnífico premio se duplican.

Si resultas ganador puedes elegir uno de estos SUPERPREMIOS:

1) UN FANTASTICO VIAJE PARA DOS PERSONAS DE 9 DIAS AL EPCOT CENTER; visitarás Marineland, el Museo Aeroespacial de la Nasa, Disnyeworld, el Epcot Center y otros muchos lugares.
2) UN SUPERORDENADOR IBM-PC portátil.
3) UNA PAGA MENSUAL DE 30.000 Pts. durante un año para ti solo.

Encontrarás las Bases para participar en el concurso, junto con las fichas, en cada programa. Envíalas a ALEA antes del 21 de Julio de 1986.



¡Atención!, si envías tus fichas antes del 23 de Junio, tus posibilidades de ganar son aún mayores.

Alea también ha pensado en los más "peques", tus hermanos de 4 a 9 años. Para ellos tenemos una serie de juegos que les ayudarán al aprendizaje de la escritura y la lectura. Comprando cualquiera de ellos, participarás automáticamente en el concurso LEXA, pudiendo llegar a conseguir una beca de estudios de hasta 500.000 Pts.

Puedes pedir tus programas llamando al teléfono de Madrid (91) 446 57 64 o bien enviándonos el cupón que hay al pie de esta página. También encontrarás los programas de la colección Logicolor en la microtienda de tu barrio, El Corte Inglés y Galerías Preciados.

Animo, por sólo 3.875 Pts. obtienes un magnífico programa y ¡hasta cuatro participaciones para el gran concurso EPCOT!

¡NO LO DEJES ESCAPAR! Prográmate ahora mismo un premio alucinante.



apartado de correos 10.048 de Madrid

GRUPO SOFT

TCA

INPUT MSX

		COMMODORE 64/128	MSX/64K	SPECTRUM 48/PLUS	AMSTRAD 464/664/6128
COLECCION LEXA	EL DUENDE				
	EL TESORO				
	EL TORREON				
	EL OASIS				
COLECCION LOGICOLOR	AUTOS LOCOS				
	MANZANAS Y GUSANOS				
	REHENES				

Deseo adquirir los siguientes programas de su biblioteca, al precio de 3.875 Pts. cada uno, más 465 Pts. IVA.
Con la compra de los programas adquiero el derecho a participar en el concurso EPCOT y/o LEXA.

Forma de pago:
- ☐ Contrarreembolso
- ☐ Con cheque adjunto a nombre de Alea, S. A.
- ☐ Con cargo a la tarjeta de crédito:
  - ☐ Visa
  - ☐ American Express
  - ☐ Dinners Club

Nombre del titular: ____

N.º de la tarjeta: ____

Válida desde __/__ hasta __/__

Firma del titular:

Nombre: ____

Dirección: ____

Tel.: ____



AÑO 1 NUMERO 2

DIRECTOR:
Alejandro Diges
COORDINADOR EDITORIAL:
Francisco de Molina
DISEÑO GRAFICO:
Tomás López
COLABORADORES:
Antonio Taratiel, Luis R. Palencia, Francisco Tórtola, Benito Román, Esther de la Cal, Ernesto del Valle, Equipo Molisoft.

INPUT MSX es una publicación juvenil de EDICIONES FORUM

GERENTE DIVISION DE REVISTAS:
Angel Sabat

PUBLICIDAD: José Real-Grupo Jota
Madrid: c/ Gral. Varela, 35, 3.º-11
Teléf. 270 47 02/03
Barcelona: Avda. de Sarriá, 11-13, 1.º
Teléf. 250 23 99
FOTOMECANICA: Ochoa, S. A.
COMPOSICION: EFCA, S. A.
IMPRESION: Edime, S. A.
Depósito legal: M. 27.884-1985
SUSCRIPCIONES: EDISA,
López de Hoyos, 141. 28002 Madrid
Telef. (91) 415 97 12
REDACCION:
Alberto Alcocer, 46, 4.º
28016 Madrid. Teléf. 250 10 00
DISTRIBUIDORA
R.B.A. PROMOTORA DE EDICIONES, S. A.
Travesera de Gracia, 56. Edificio Odiseus.
08006 Barcelona
El precio será el mismo para Canarias que para la Península y en él irá incluida la sobretasa aérea.

Se ha solicitado el control OJD

INPUT MSX es independiente y no está vinculada a MSX Research o sus distribuidores.

INPUT no mantiene correspondencia con sus lectores, si bien la recibe, no responsabilizándose de su pérdida o extravío. Las respuestas se canalizarán a través de las secciones adecuadas en estas páginas.

INPUT MSX

SUMARIO

UN MUNDO MUY MOVIDO

Aunque eran de sobra conocidas las dificultades financieras por las que venía pasando durante el último año **Sinclair Research,** a todos sorprendió la noticia de la compra de ésta firma por parte de su más directo rival: **Amstrad.** Se da la paradoja de que además la firma fue adquirida a precio de saldo, sin contar las deudas contraídas con los proveedores.

Alan Sugar es el artífice del *boom* de **Amstrad,** un hombre conocedor de los mercados internacionales de componentes, en los que se mueve ágilmente desde que lanzase sus primeros equipos de alta fidelidad. Ambos personajes tiene algo más en común que ser británicos, comparten la idea fija de poner los avances de la tecnología electrónica al alcance de cualquier consumidor.

Sin embargo, mientras que *sir* **Clive** tiene vocación investigadora, **Sugar** es un experto en el control de compras y *stocks,* factor este último que llevo a **Sinclair** a la bancarrota, al acumularse una gran cantidad de equipos invendidos en sus almacenes. Es curioso observar como la compañía que no ha mucho estuvo valorada en 136 millones de libras esterlinas, haya sido comprada por tan sólo 5.

Por otro lado ambas firmas parece que disponen de modelos muy avanzados de máquinas dispuestas a enfrentarse en las tiendas con los **Amiga,** de **Commodore,** y **520 ST,** de **Atari** e incluso los atractivos **MSX II.** Este tipo de competencia tan directa promete ser estimulante para el mercado y la producción de nuevo *software.* Pero sin duda los más beneficiados deberán ser los usuarios que accedan a la nueva generación de microordenadores con elevadas prestaciones.

Algunos de los que se incorporan a nuestro inmejorable conjunto de lectores, escriben a la redacción para expresarnos su desconcierto por la paginación del artículo que ocupa las ocho páginas centrales de INPUT. Piensan que su ejemplar está mal encuadernado o que nos hemos confundido al hacer la revista. Nada más lejos de la verdad para su tranquilidad. Se trata de un curso de aprendizaje del BASIC —código máquina en el futuro— a través de un coleccionable, que explica el modo de desarrollar juegos propios. Dentro de algunos números lanzaremos unas tapas que permitirán que este coleccionable se transforme en un libro y, por tanto, cada capítulo no tiene porqué ocupar las ocho páginas, pudiendo ser más o menos.

LOS MEJORES DE INPUT

Hemos pensado que es interesante disponer de un ***ranking*** que ponga en claro, mes a mes, cuáles son los programas preferidos de nuestros lectores. Para ello, es obligado preguntaros directamente y tener así el mejor termómetro para conocer vuestras preferencias. Podéis votar por cualquier programa aunque no haya sido comentado todavía en **INPUT**.

El resultado de las votaciones será publicado en cada número de **INPUT**.

Entre los votantes sortearemos 10 cintas de los títulos que pidáis en vuestros cupones.

Nota: No es preciso que cortéis la revista, una copia hecha a máquina o una simple fotocopia sirven.

Enviad vuestros votos a: **LOS MEJORES DE INPUT** Alberto Alcocer, 46 - 4.° B. 28016 Madrid

ELIGE TUS PROGRAMAS

Primer título elegido ____________________

Segundo título elegido ____________________

Tercer título elegido ____________________

Programa que te gustaría conseguir ____________________

Qué ordenador tienes ____________________

Nombre ____________________

1.er Apellido ____________________

2.° Apellido ____________________

Fecha de nacimiento ____________________

Teléfono ____________________

Dirección ____________________

Localidad ____________________

Provincia ____________________



HAN LLEGADO LOS MSX2



Una resolución gráfica de 512x212 puntos en 16 colores, otra de 256x212 en 256 colores (de una paleta de 512), una memoria de vídeo de 128K, un reloj en tiempo real alimentado permanetemente y una unidad de diskettes de 3.5 pulgadas incorporada, son algunas de las características de los nuevos MSX2, que podremos ver en las tiendas, probablemente, entre los meses de Mayo y Junio.
El nivel de precios de estas máquinas es francamente interesante. Philips sitúa el de su modelo VG-8235 (foto derecha) en torno a las 130.000 ptas. incluido el monitor. El HB-F500P de Sony, (a la izquierda) que incluye una unidad de diskettes de 1Mbyte, rondará las 145.000 ptas.





MUCHOS Y BIEN AVENIDOS

Como muy bien dice el refrán: la unión hace la fuerza.
Con esta filosofía se ha celebrado, durante la primera semana del mes de Abril, en Barcelona, una reunión entre las diferentes marcas que comercializan equipos dentro del estandar MSX.
En dicha reunión, en la que estuvieron presentes Canon, Dynadata, Mitsubishi, Panasonic, Philips, Pioneer, Sanyo, Sony y Toshiba, se formalizó un primer acuerdo de colaboración.
En dicho acuerdo se definieron una serie de aspectos comunes a todas la marcas, sobre los que actuar de forma conjunta. Entre ellos cabe destacar:
La difusión del MSX como primer estandar, la difusión del actual catálogo de *software* (cerca de 400 títulos), la participación conjunta en ferias y la difusión de noticias y artículos referentes al estandar.
Varias de las marcas han expresado su confianza en los resultados, a corto plazo, de este acuerdo.



MAS CONFORT EN LA PANTALLA

Essilor es una importante firma francesa de la industria óptica. Durante algún tiempo sus técnicos han estado desarrollando lo que han dado en definir como "una lente antichoque, dos veces más ligera que la lente mineral".
Este producto comercial, Essilor Soft, ha sido desarrolado para proteger la vista de quienes trabajan habitualmente con monitores de vídeo.
Su función consiste en evitar el deslumbramiento y los reflejos, aumentando asimismo el contraste de la imagen.
Para presentar este avance de la óptica, fue convocado un panel de expertos en distintas áreas que ofrecieron una panorámica más clara de los problemas visuales derivados del trabajo con pantallas conectadas a ordenador y sus posibles compensaciones.



AMSTRAD SIGUE IMPACTANDO.

Imparable, Amstrad sigue ganado puestos, hasta tal punto que se estima puedan haberse vendido casi cincuenta mil unidades del CPC 6128, previéndose un fuerte incremento en base al menor P.V.P., unas ochenta y cinco mil pts. en la versión con fósforo verde y cerca de ciento veinte mil si es de color, IVA aparte, por supuesto.
Otra baza que parece estar jugando este fabricante es la del sistema operativo CP/M, de Digital Research, que aún no siendo precisamente de nuevo cuño, sí dispone de una gran biblioteca de programas. Este mismo sistema está siendo utilizado por Commodore en el C-128, aunque ya existía en cartucho para el C-64, y se especula en torno a que Sinclair lo utilizara en breve (curiosa coincidencia con la reciente absorción de Sinclair).
Una de las dificultades que surgen con el CP/M es la falta de compatibilidad con el formato del *diskette*. A este respecto, el CPC 6128 dispone ya de programas profesionales tan populares como el Multiplan (potente hoja de trabajo), el SuperCalc II (otra popular hoja), el dBase II (una de la bases de datos más extendidas), etc..

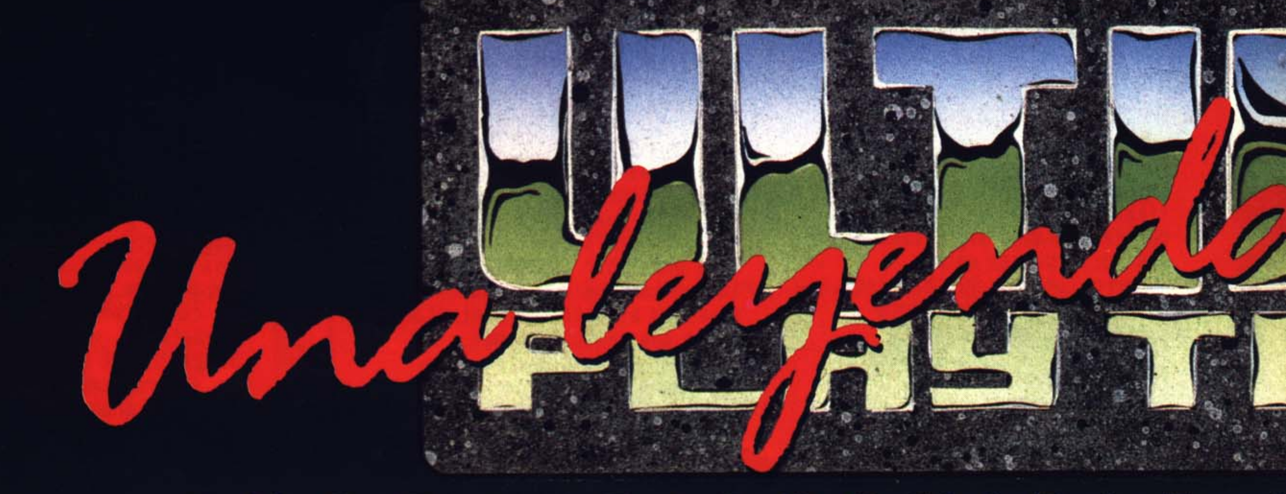
SI BUSCAS LO MEJOR



■ ALIEN 8 ■



■ KNIGHT LORE ■

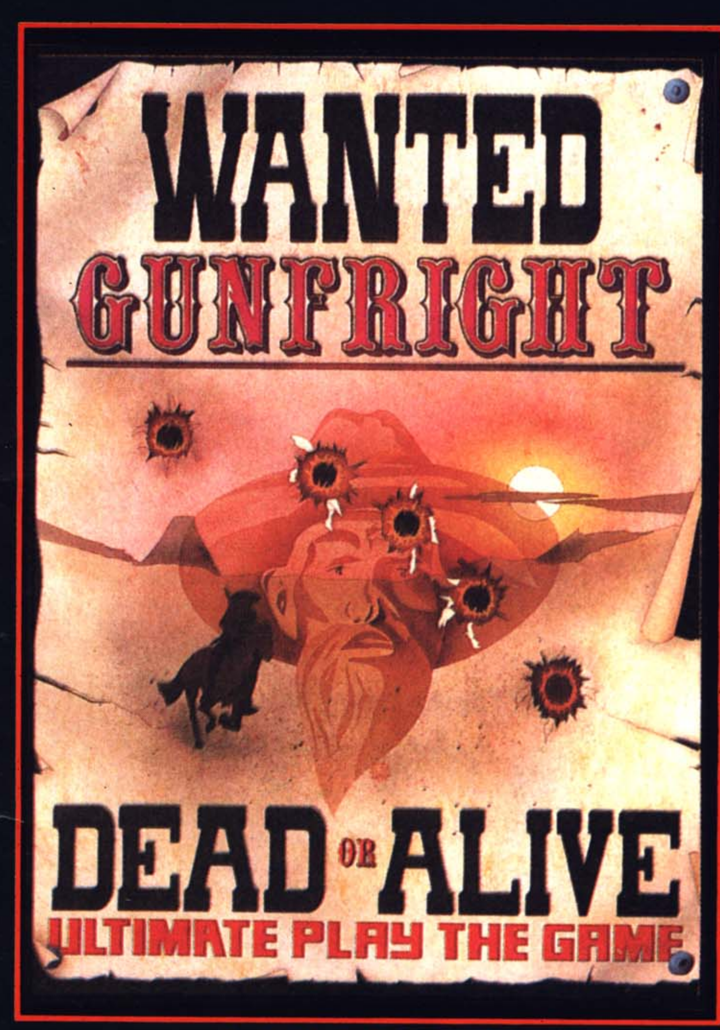


AHORA DISPONIBLE

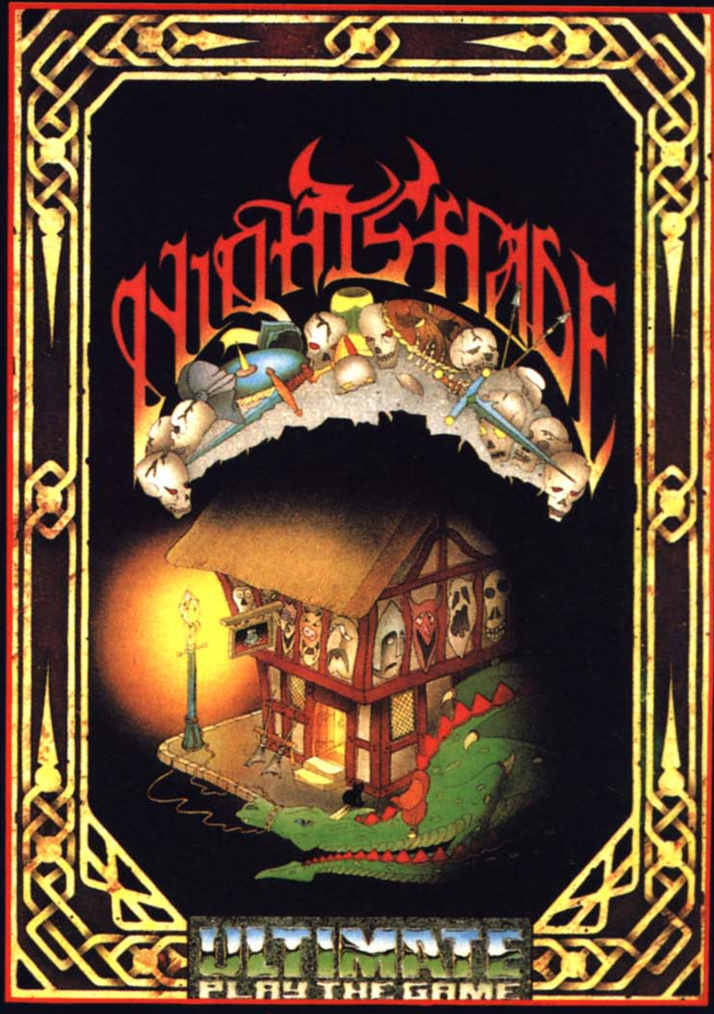
DISTRIBUIDOR EXCLUSIVO PARA ESPAÑA, ERBE SOFTWARE, SANTA ENGRACIA

Software









NIGHTSHADE



ARA TODOS LOS «MSX»

el: 447 34 10. DELEGACION BARCELONA, Avd. MISTRAL, 10. Tel. (93) 432 07 31



TOSHIBA EN LA BRECHA

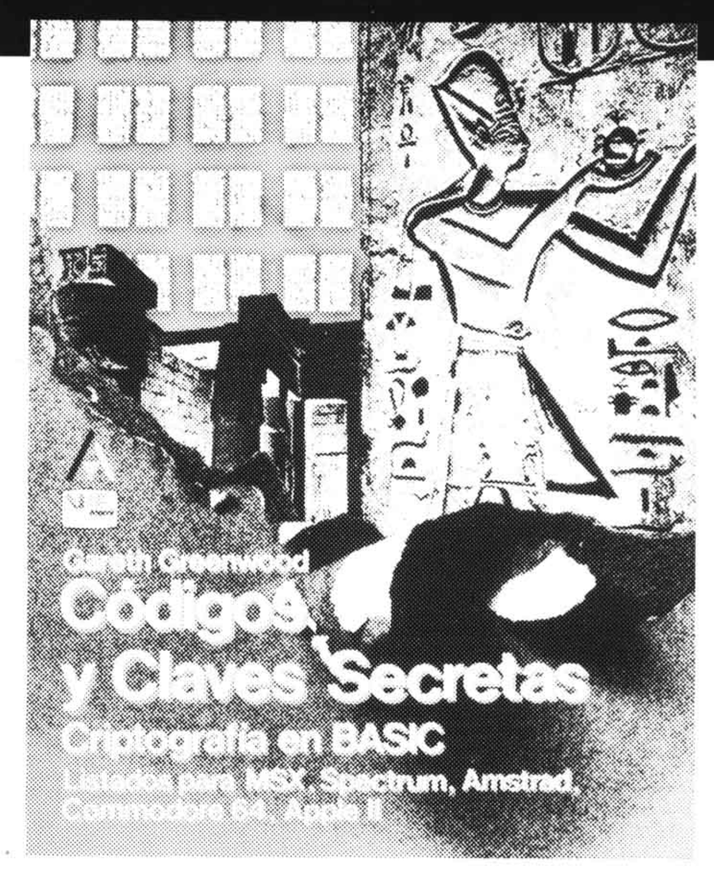
Los musicos y educadores estan de enhorabuena, Toshiba tiene algo para ellos. Los primeros disponen ya de un teclado que convierte al MSX en un verdadero sintetizador. Los segundos cuentan con un Aula Informática, capaz de acometer la tarea de auxilio en la enseñanza de diversas materias que van desde las Matemáticas al Ingles.



NEW MEDIA SYSTEMS

Se trata un nuevo concepto, propugnado por las firmas de electrónica de consumo acogidas al estandar MSX. Aunque con miras muy amplias pero poco definidas todavía, el concepto de NMS buscará, en principio, la comunicación de un ordenador central (un MSX2) con sistemas de vídeo, cadenas de audio, sistemas *compact disc*, otros ordenadores y en general con cualquier sistema electrónico de consumo que se haya vendido o pueda llegar a venderse de forma masiva.

El desarrollo del concepto vendrá de la estandarización de formatos y conexiones entre los productos desarrollados por las diversas firmas, lo que permitirá conectar cualquier cosa a cualquier otra. Esta es la dirección en la que actualmente se trabaja y mediante la que se buscarán nuevas vias de comercialización no sólo de ordenadores, sino también de una amplia gama de productos de electrónica de consumo.



Ordena tus propias ideas.

Le sacarás partido a tu ordenador



Adquiéralos en su librería habitual. Si no le es posible o desea que le enviemos nuestro catálogo, envíe este cupón a: *Apdo. de Correos 14632*
Ref. D. de C.
28080 MADRID

☐ Les ruego me envíen el catálogo de su editorial.
☐ Les ruego me envíen los siguientes títulos:

____

____

TOTAL ________

☐ Adjunto talón bancario a
GRUPO DISTRIBUIDOR EDITORIAL, S. A.

☐ Pagaré contrarrembolso (+ 125 pesetas de gasto de envío).

☐ Giro postal.

Nombre ________________

Profesión ________________

Dirección ________________

C. P. ________ Localidad ________

Provincia ________________

IM

Códigos y Claves Secretas: Criptografía BASIC

Gareth Greenwood

* *1.378 ptas.*

La criptografía y el criptoanálisis ha jugado papeles muy importantes en muchos acontecimientos políticos y diplomáticos de la historia.
CODIGOS Y CLAVES SECRETAS: CRIPTOGRAFIA EN BASIC es una introducción a la criptología para usuarios de microordenadores. A lo largo del libro se describen y analizan las distintas técnicas de cifrado y descifrado de mensajes y se acompañan los programas en BASIC para que el ordenador realice el trabajo repetitivo a partir del ingenio, intuición y destreza del "rompeclaves".
Con tu ingenio y este libro aprenderás técnicas útiles, te divertirás, desarrollarás tu mente y, quién sabe, ¡tal vez descubras algo que antes no sabías!
El libro contiene versiones completas de todos los programas para SPECTRUM, AMSTRAD, MSX, COMMODORE y APPLE II.

Simulaciones: Replica la realidad con tu ordenador

Tim Hartnell

* *1.643 ptas.*

Cómo desarrollar y disfrutar con las simulaciones por ordenador. Contiene 18 programas de simulación, listos para funcionar, que incluyen: un simulador de vuelo, una tortuga robot y la máquina de ajedrez de Torres Quevedo.
Incluye versiones para SPECTRUM, AMSTRAD, MSX, COMMODOROE y APPLE II

Sistemas Expertos: Introducción al diseño y aplicaciones

Tim Hartnell

* *2.120 ptas.*

Explora el fascinante mundo de los Sistemas Expertos en tu microordenador. Entre otros programas, este libro contiene dos interesantes intérpretes de PROLOG y LISP realizados en BASIC.
Incluye versiones para SPECTRUM, AMSTRAD, MSX, COMMODORE y APPLE II

Otros títulos:

Descubre tu MSX.
(Programación y aplicaciones)
Joe Pritchard
* *1.272 ptas.*

El libro gigante de los juegos para MSX.
Andrew Lacey
* *1.590 ptas.*

MSX: Guía del programador y manual de referencia.
T. Sato, P. Mapstone e I. Muriel
* *2.279 ptas.*

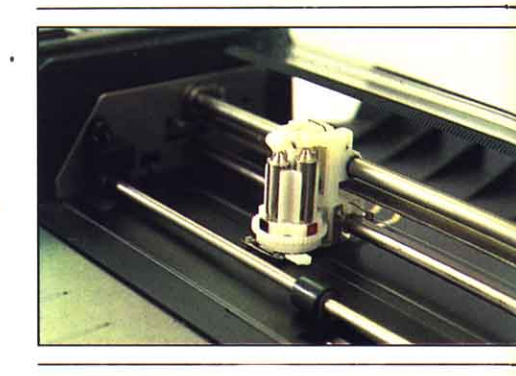
Lenguaje máquina para MSX.
(Introducción y conceptos avanzados)
Joe Pritchard
* *1.537 ptas.*

Cometas en tu micro: El Halley.
(Cálculos de órbitas y parámetros de cometas en BASIC) F. Galende Domínguez, A. Sánchez López, M. Alparaz López y J. A. Sánchez García
* *583 ptas.*

* IVA incluido

ANAYA ANAYA ANAYA ANAYA ANAYA ANAYA

MI ORDENADOR YA PUEDE DIBUJAR



El dispositivo electromecánico que permite traducir los datos proporcionados por un ordenador en un dibujo recibe el nombre de *plotter*, o trazador. Una plumilla se desplaza —bajo el control de un programa— por toda la superficie de una hoja de papel, dejando tras de sí un trazo impreso.

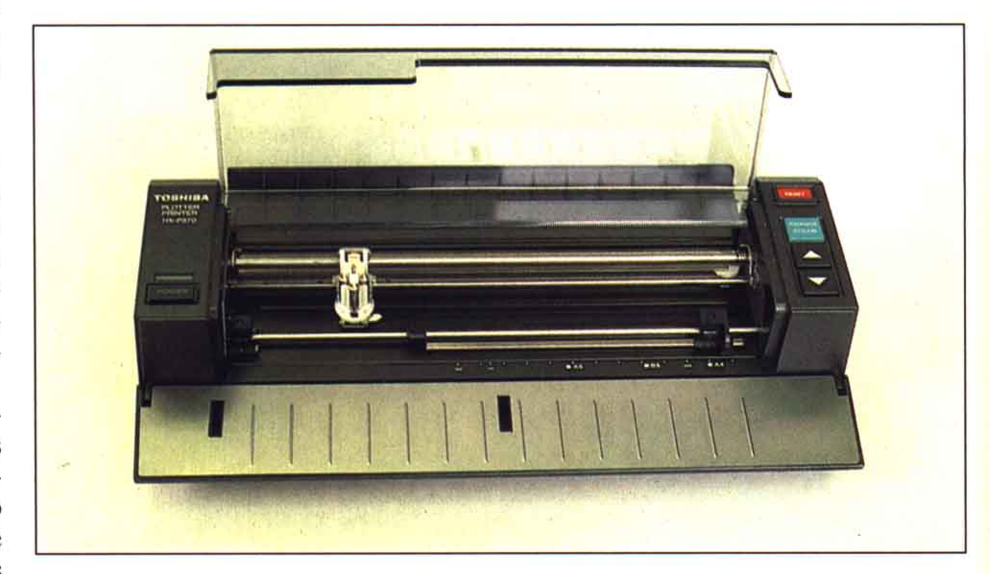
Una máquina capaz de dibujar automáticamente fue uno de los grandes sueños que acompañaron al hombre desde la antigüedad. Se elaboraron muchos modelos mecánicos de aplicación limitada, pero no es hasta la llegada del ordenador que ésta aspiración comienza a materializarse.

El *plotter*, por tratarse de una herramienta construida en base a piezas mecánicas de precisión, es un atractivo periférico situado lejos del bolsillo de un usuario doméstico. Hasta hace poco no era posible encontrarlos más que en los laboratorios y empresas importantes. Un reducido grupo de fabricantes japoneses ha desarrollado algunos dispositivos compactos que posibilitan la fabricación de *plotters* de pequeño formato y precio adsequible.

Antes de entrar en detalle veamos cómo son los *plotters* estándar, que se agrupan en dos familias genéricas y, después, entraremos con el modelo **HX-P570**, de **Toshiba**.

La primera familia ha sido bautizada como *plotters* de mesa. En efecto, su apariencia es la de una mesa de formato variable (normalmente DIN A0, A1, etc.) en cuya superficie reposa la hoja de papel. La plumilla de dibujo puede deslizarse a través de una barra que cruza la mesa de un lado a otro. A su vez, esta barra puede desplazarse en dirección perpendicular a sí misma. De ese modo la plumilla dispone de los dos grados de libertad necesarios para situarse en cualquier punto del papel, definido por su ordenada y abcisa. El tercer movimiento posible consiste en la elevación de la plumilla.

El ordenador puede controlar las evoluciones de la plumilla, decidiendo en que momentos la tinta entra en





contacto con el papel y, por tanto, dibujar desde los perfíles del ala de un reactor hasta una proyección tridimensional de un *chalet*, incluso figuras matemáticas. Todo depende de que se coordinen adecuadamente los movimientos en los sentidos X e Y.

La segunda familia no es más que una variante de la anterior, sólo que la base plana es sustituida por un tambor sobre el que descansa el papel. La plumilla se desplaza igualmente en sentido longitudinal por una barra, cuya dirección es paralela a la del eje del tambor. Esto unido a su giro y la elevación de la plumilla, hacen que este sistema sea igualmente útil para el dibujo.

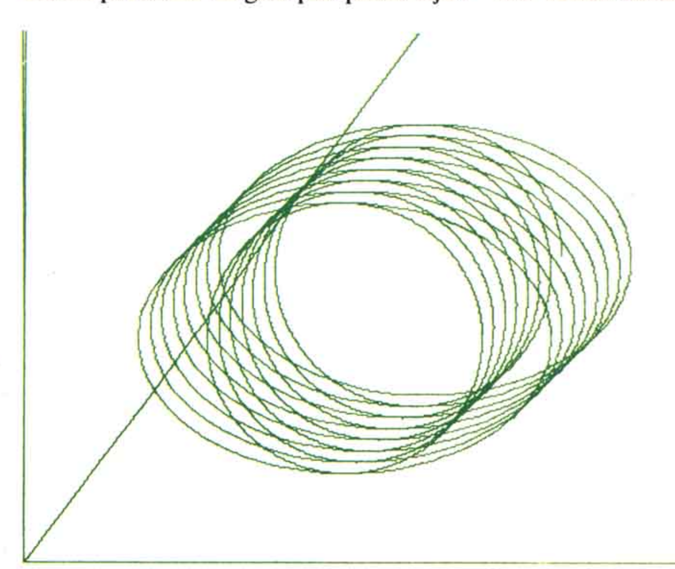
Existen distintas alternativas para la plumilla, desde un simple bolígrafo hasta dispositivos que utilizan *toner* (un líquido similar al empleado en las fotocopiadoras) y un dispositivo para secar rápidamente la tinta.

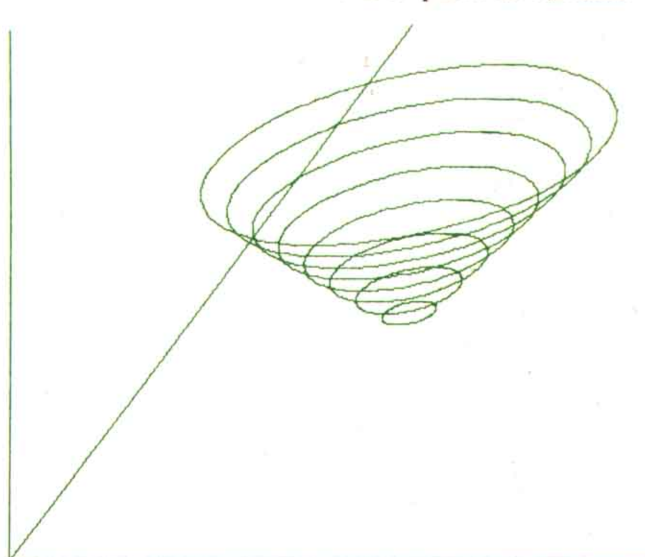
Las crecientes necesidades de las empresas y la aparición de *software* para gráficos comerciales han forzado la aparición de *plotters* más asequibles, aunque los dibujos no sean de elevada perfección. Uno de los factores que inciden con mas fuerza en el precio final de un *plotter* es la resolución, en este caso la distancia entre dos sucesivos pasos en el avance de la plumilla.

El *plotter* de **Toshiba** tiene características comunes a ambas familias. Aquí la barra por la que desliza el cabezal de dibujo no se desplaza, es el papel quien lo hace con la ayuda de

dos rodillos dentados que muerden los laterales del papel. La plumilla es sustituida por un revólver que alberga cuatro diminutos bolígrafos con tintas de distintos colores.

El ordenador envía sencillas órdenes en forma de cadenas de datos (ASCII) a través del *port* **Centronics**. La electrónica interna del *plotter* los convierte en impulsos de control que activan a los motores paso a paso, capaces de producir un giro por pasos fijos

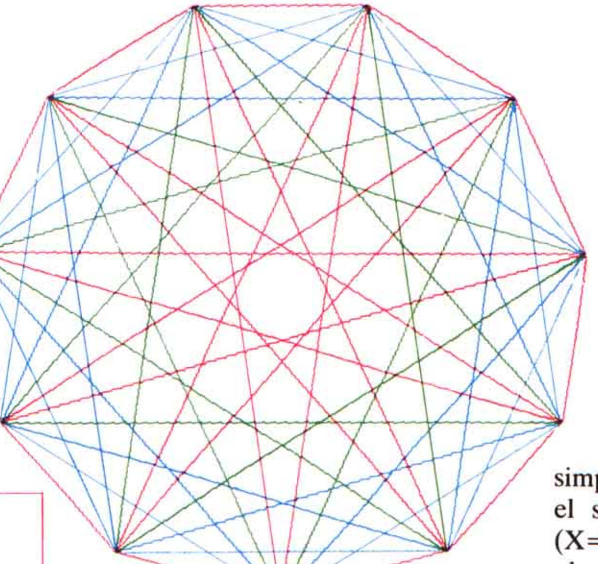


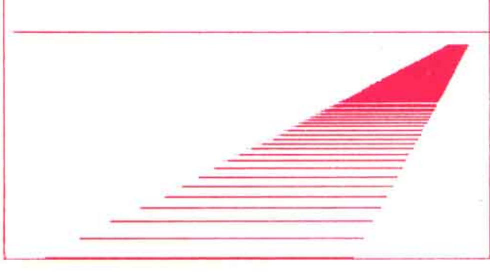


menores que una revolución completa de su eje. Un conjunto de engranajes reduce la amplitud de este movimiento, que aplicado a los rodillos que desplazan el papel o la plumilla, hacen que los avances en ambas direcciones se produzcan en incrementos de 0,2 milímetros.

El **HX-P570** es modelo extremadamente compacto y de apariencia robusta, que no muestra más que un par de conectores. El primero está destinado al cable que procede del transformador de alimentación y el segundo es del tipo habitualmente utilizado en el estándar **Centronics**, por el que se conecta directamente al ordenador, sin precisar nada más.

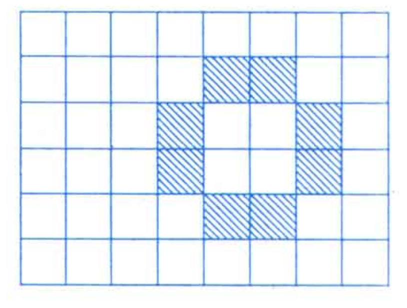
Las dimensiones físicas son de 310 mm. de ancho por 67 de alto y 108 de fondo, pero las caras laterales son abatibles, de tal forma que durante su utilización se convierten en una superficie para soportar el papel, que pueden ser folios estándar de formatos DIN A4, B5, A5 o incluso el clásico rollo de papel suministrado con el equipo.





En la parte superior izquierda aparece el botón de alimentación, junto a un LED (diodo emisor de luz) verde, que indica cuando está recibiendo alimentación eléctrica. En el lado izquierdo hay cuatro pulsadores, que corresponden a la inicialización (*reset*), el cambio de bolígrafo (color) y avance del papel en ambos sentidos.

El número máximo de pasos admisibles en el sentido horizontal (eje X) va desde 410, en el caso del rollo de papel de 10 cm. de anchura, hasta los 960 pasos para el DIN A4. En la dirección vertical (eje Y) este margen alcanza los 1384 pasos de 0,2 mm.

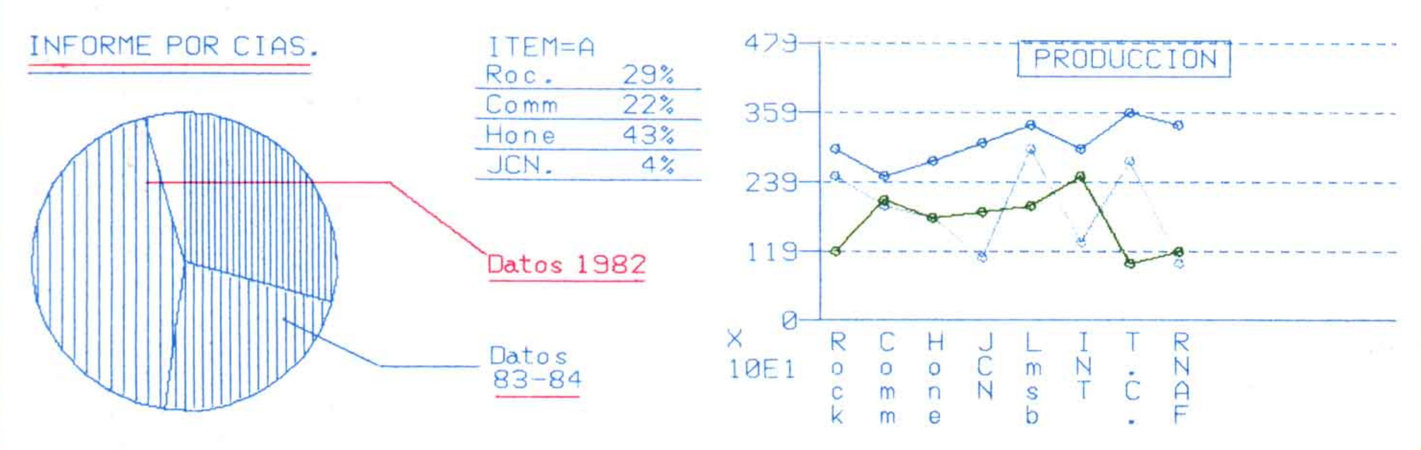


El *plotter* dispone de un conjunto de comandos elementales que cubren las necesidades para el desarrollo de dibujos. El envío de la letra I, de sintáxis tan simple como: LPRINT «I», inicializa el sistema, definiendo como origen (X=0, Y=0) la posición actual de la plumilla. H envía a la plumilla a la posición origen en cualquier momento. Por su lado C seguida de un número entre el 0 y el 3 cambia el color del trazo al del bolígrafo de igual número. DX1,Y1 realiza un trazo desde la posición en que esté la plumilla hasta el punto de coordenadas (X1,Y1), pero

T-GRAPH DIBUJA POR TI

Una de las aplicaciones mas tradicionales de las impresoras *plotter* ha sido siempre la representación gráfica de datos numéricos. Nos referimos a los conocidos gráficos de barras, histogramas, gráficos de tarta, etc.

El uso de este tipo de gráficos está ampliamente difundido y los mismos gozan de bastante prestigio. No es para menos. Cumplen una función de gran importancia: ahorrar tiempo a la hora de la comprensión e interpretación de largas



y áridas series de datos numéricos.

La impresora *plotter* es el periférico ideal para la creación de este tipo de gráficos. Es capaz de representarlos con mejor calidad que cualquier impresora matricial y además, por la propia filosofía de las impresoras *plotter*, pensadas sobre todo para el diseño de gráficos, resulta mucho más sencillo escribir para ellas el programa necesario para dibujar.

Aún así, un programa que tenga en cuenta todos o casi todos los factores que intervienen a la hora de transferir la información de una tabla de datos a, por ejemplo, un gráfico de tarta, (factores como la escala, el cálculo de los porcentajes, la selección del tamaño de los dibujos, el envío de comandos al *plotter*) es algo que queda fuera del alcance del usuario medio. Por ello nos imaginamos que habrá más de uno que se alegre enormemente al conocer, a través de este comentario, al programa **T-GRAPH**.

GRAFICOS DE TODO TIPO

T-GRAPH es un programa desarrollado por **Toshiba** y pensado para su utilización junto con la impresora *plotter* **HX-P570**. Claro está, el estándar **MSX** permite utilizarlo con cualquier ordenador **MSX**. El programa se encarga de todas las labores comprendidas entre la entrada de datos por parte del usuario y el momento en el que el mismo usuario retira la hoja en la que ha quedado impreso el gráfico. Con otras palabras, **T-GRAPH** se encarga prácticamente de todo.

El programa se maneja desde un menú del que se seleccionan las diversas opciones al pulsar las teclas de función. Vamos a ver cuáles son las principales de estas opciones y para ello seguiremos el camino normal de un usuario que quiera obtener un gráfico.

En primer lugar habrá que escoger la opción de entrada de datos. La misma permite introducir valores numéricos para un total de hasta cuatro conceptos diferentes (por ejemplo los datos de 4 compañías, o las notas de los alumnos de 4 cursos diferentes). Por cada concepto se pueden introducir hasta 12 valores númericos distintos (todos ellos enteros entre 1 y 99999999). La opción de entrada de datos se encarga de calcular automáticamente el valor total y el valor medio correspondientes a cada uno de los 12 valores, según los cuatro conceptos. También es posible llevar a cabo una ordenación según cualquiera de los cuatro conjuntos de valores.

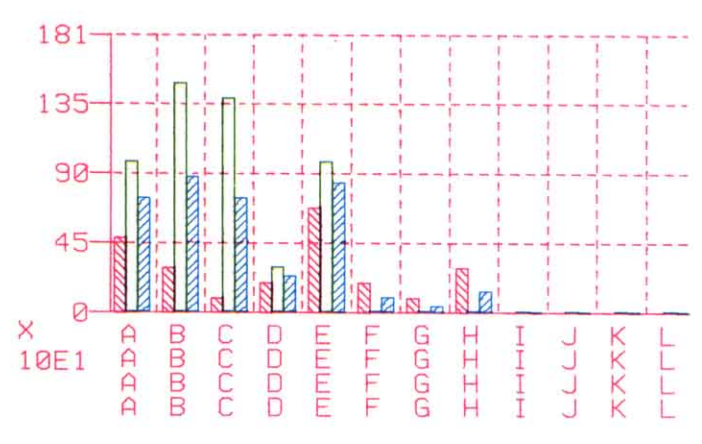
si a estas dos cifras les siguen otras la plumilla sigue su avance. Por ejemplo, LPRINT «D23,45,78,132» traza una línea desde el punto actual hasta (23,45) y desde este punto dibuja hasta el (78,132). Añadiendo nuevas coordenadas continúa dibujando nuevas líneas.

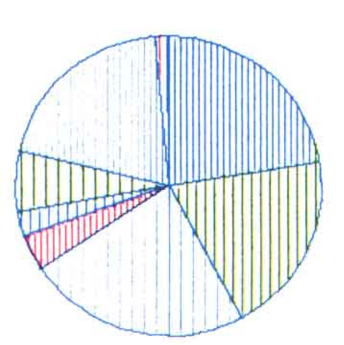
El comando M hace lo mismo que D, sólo que la plumilla no toca el papel, simplemente se desplaza.

También son posibles movimientos relativos, esto es que la última posición actúa como referencia, así LPRINT «J23,36» dibuja una línea hasta el punto situado 23 pasos más allá en la dirección X y 36 pasos en la Y. Por otro lado, R puede compararse a M, sólo que la plumilla realiza un movimiento relativo pero sin dibujar.

Quienes conozcan el lenguaje LOGO verán en estos comandos una gran similitud con los movimientos de la tortuga definibles por programa.

El *plotter* **HX-P570** actúa también como impresora. En este caso los doscientos cincuenta y dos caracteres y





ITEM=T	
EEEE	22%
AAAA	19%
BBBB	23%
HHHH	3%
FFFF	2%
DDDD	6%
CCCC	19%
GGGG	1%
IIII	0%
JJJJ	0%
KKKK	0%
LLLL	0%

Una vez introducidos los datos, que pueden modificarse en cualquier momento, se puede pasar a la fase de ejecución del gráfico. En primer lugar habrá que escoger entre los cuatro tipos de gráficos posibles: gráfico de barras, lineal, de tarta o de bandas. A continuación habrá que decidir si queremos el gráfico en tamaño pequeño o grande, si queremos el interior del gráfico rayado o sin rayar y en cual de los cuatro colores que ofrece el *plotter* deseamos ver nuestro gráfico. Hecha esta selección (es inmediata, sin más que pulsar tres o cuatro veces algunas teclas de función) podemos pasar a ver cómo queda el gráfico, primero en la pantalla del televisor y a continuación, si estamos satisfechos, ya sobre el papel a través de la impresora *plotter* o bien a través de una impresora matricial (en este último caso lo que obtenemos es un *hard-copy* de la pantalla en blanco y negro y sin demasiada calidad).

En caso de que queramos consevar nuestro gráfico, podemos utilizar la opción para almacenarlo en *cassette*.

Pero no acaban aquí las posibilidades de este magnífico programa.

Supongamos que queremos añadirle a nuestro gráfico más información, pero información de otro tipo distinto. Por ejemplo, al lado de nuestro gráfico de barras puede que deseemos varias líneas que, saliendo del gráfico, incluyan textos explicativos, o tal vez queramos darle un título a nuestro gráfico y subrayarlo en un color diferente, o hasta puede que se nos ocurra incluir un mapa de las regiones a las que hace referencia el gráfico. Todo este tipo de geniales ideas, que sin duda se nos ocurrirán, pueden llevarse a cabo desde una opción de dibujo presente en el programa. Esta opción permite dibujar cualquier cosa en la misma pantalla que nuestro gráfico y por supuesto también a través de la impresora *plotter*. Además, y esto resulta bastante curioso, nuestro gráfico y los dibujos que hemos añadido, son independientes, es decir que podemos escoger otro tipo de gráfico e incluirle los mismos dibujos adicionales sin necesidad de tener que hacerlos de nuevo.

TODO ES AUTOMATICO

Lo más interesante del programa es que realiza todos los ajustes necesarios para dibujar el gráfico de forma autónoma, sin que el usuario tenga que intervenir de ninguna forma. Esto quiere decir que el usuario no tiene necesidad de saber nada sobre el *plotter*, ni sobre como se programa para dibujar líneas, ni sobre cuales son los comandos de cambio de color, ni cómo conseguir un cambio de escala.

Todo, absolutamente todo, lo hace **T-GRAPH**. Lo único que se le pide al usuario es que introduzca los datos a representar y que decida cómo quiere el gráfico.

No lo habíamos mencionado, pero existe la posibilidad de escoger la escala de forma manual o bien de forma automática y también se puede decidir si van a aparecer divisiones en los gráficos o sólo el gráfico en sí.

En cuanto a la calidad de los resultados, preferimos que cada uno juzgue por sí mismo. No hay más que echar un vistazo a los gráficos que acompañan a este comentario.

símbolos gráficos son dibujados en dieciséis distintos tamaños, en formatos que van desde 0,8 por 1,2 mm. hasta 12,8 por 19,2 mm.

La velocidad media de impresión es de 6 caracteres por segundo, pudiendo obtenerse hasta 160 caracteres en una línea de papel DIN A4.

Los caracteres pueden dibujarse en las cuatro orientaciones posibles, hacia arriba o abajo y derecha e izquierda.

Las aplicaciones de un *plotter* son infinitas, abarcando campos que van desde la educación hasta los gráficos comerciales, sin olvidar sus posibilidades artísticas, como muestran las ilustraciones y programas de ejemplo que acompañan a este artículo.

```
5 REM**VOLCADO DE CARACTERES
  DIBUJADOR POR EL USUARIO
  EN FORMATO GRANDE
10 LPRINT CHR$(27)+"#":REM**
   PASO A MODO GRAFICO
20 LPRINT "MO,-400"
30 LPRINT "I"
40 FOR I=0 TO 8
45 REM**DIBUJO DE LA REJILLA
```

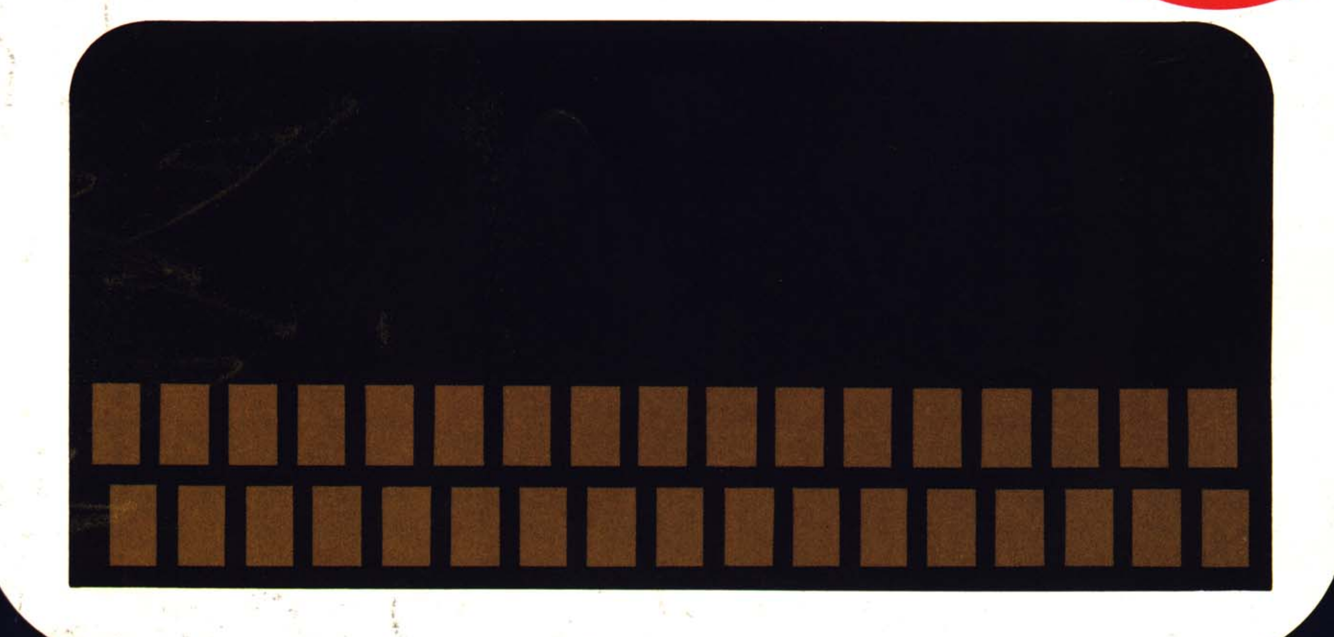
NO ESTAMOS EN EL AÑO 2000 PERO SERMA TE PRESENTA EL NUEVO SOPORTE DEL FUTURO PARA ALMACENAR DATOS LA TARJETA

SOFTCARD

16 K, 32 K, 64 K, 256 K.



PESO Y DIMENSIONES COMO UNA TARJETA DE CREDITO



YA A LA VENTA
TUS PRIMEROS JUEGOS
EN TARJETA






1X2



LE MANS 2



QH

1. 

2. 

TARJETA SOFTCARD

1. FORMATO PRESENTACION
2. ADAPTADOR PARA «MSX»

CON LA COMPRA DE LAS TARJETAS «1 X 2» ó «QH» ADAPTADOR GRATIS

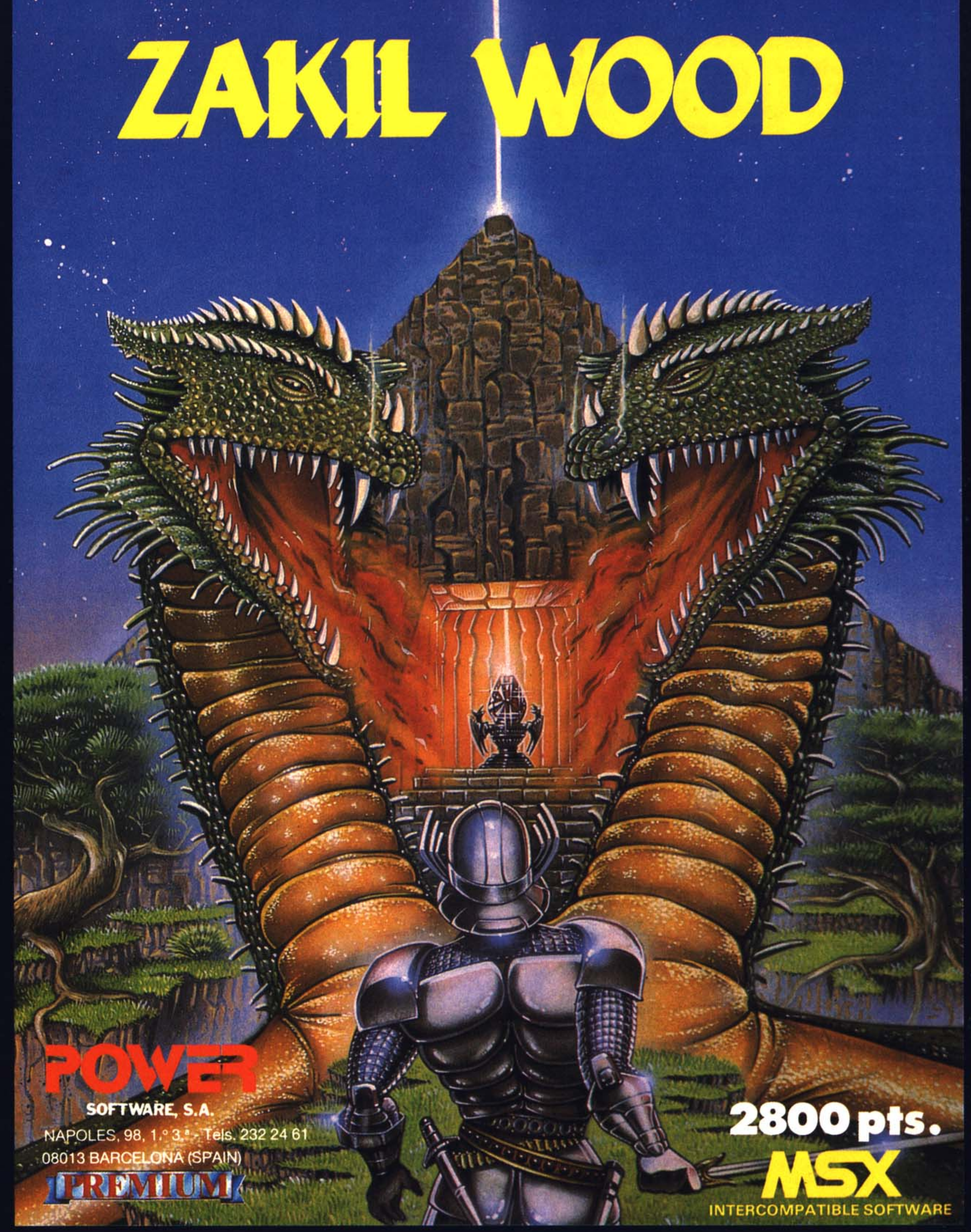
A LA VENTA EN TODOS LOS ESTABLECIMIENTOS DE GALERIAS

Y EN TODOS LOS DISTRIBUIDORES DE NUESTROS PRODUCTOS

```
50 LPRINT "M90,";I*30+30
60 LPRINT "D270,";I*30+30:IF
   I>6 THEN 90
70 LPRINT "M";I*30+90;"
   ,30"
80 LPRINT "D";I*30+90;"
   ,270"
90 NEXT
100 LPRINT"Q3"
110 FOR I=5 TO 0 STEP-1
120 X=270-I*30-5:Y=275
130 LPRINT "M";X",";Y
140 LPRINT "P";2^I
150 NEXT:LPRINT "Q0"
160 FOR I=1 TO 8
170 READ P:Q=P
180 FOR J=5 TO 0 STEP-1
190 PP=INT(2^J+.1)
200 IF P>=PP THEN P=P-PP:
    GOSUB 400
210 NEXT J
220 X=290:Y=270-I*30+5
230 LPRINT "M";X",";Y
240 LPRINT "P";
250 IF Q<100THEN LPRINT " ";
260 IF Q<10 THEN LPRINT," ";
270 LPRINT " ";Q
280 NEXT
400 X=30*(8-J):Y=270-30*I
410 FOR A=0 TO 30 STEP 5
420 LPRINT "M";X",";Y+A
430 LPRINT"D";X+30-A;",";
    Y+30
440 NEXT
450 FOR A=5 TO 30 STEP 5
460 LPRINT "M";X+A",";Y
470 LPRINT "D";X+30;",";
    Y+30-A
480 NEXT
490 RETURN
500 DATA 28,13,45,11,4,23,52
    ,56
```



¡LA AVENTURA MAS EMOCIONANTE!



ZAKIL WOOD

Deseo recibir los juegos que a continuación especifico, comprometiéndome al pago del importe de los mismos.

Nombre ______________________

Dirección ______________________

Teléfono ______________________

Firma:

SISTEMA ______________ CANTIDAD ______________

☐ Contrarreembolso. ☐ Adjunto Talón. ☐ Giro Postal.

Deseo recibir información de sus programas en: MSX ☐ AMSTRAD ☐

ENVIA MENSAJES SECRETOS (y II)

- RUPTURA DE CODIGO
- FRECUENCIA
- NUEVOS METODOS DE CODIFICACION
- DICCIONARIOS DE CODIGOS

Aquí tienes algunos procedimientos más sutiles para enviar información secreta, preservándola de caer en manos inadecuadas. También se examinan brevemente algunos métodos de ruptura de códigos.

Como vimos en el artículo anterior (en el número 1 de **INPUT MSX**) el problema de la codificación de la información reservada admite muchos planteamientos. Algunos de estos métodos son relativamente fáciles de descifrar, pero con ayuda de los ordenadores es posible emplear procedimientos cada vez más complicados.

RUPTURA DE CODIGOS

A medida que los criptógrafos se esfuerzan en inventar códigos mejores y más seguros se produce un esfuerzo paralelo por parte de los «rompedores de códigos», que hacen todo lo posible por frustar el esfuerzo de los primeros.

En los casos más sencillos de cifrado por **transposición** o **sustitución** se dispone de una potente herramienta, consistente en contar la frecuencia con la que aparecen todas las letras.

Cada idioma dispone de una serie de letras que aparecen con más frecuencia en un texto cualquiera que otras, en castellano éste es el caso de las E, A, O, N, S, I. Según esto, si después de contar el número de veces que aparece cada letra en un texto cifrado siguen siendo las más frecuentes EAONSI, lo más probable es que te encuentres ante un cifrado por **transposición**. Si, por el contrario, son otras las letras que aparecen con mayor frecuencia, deberás considerar la posibilidad de que se trata de un cifrado por **sustitución**.

En ambos resulta necesario contar la frecuencia de cada letra en el texto del mensaje. Esta es una labor que manualmente ejecutada requiere tiempo y está sujeta a errores. El programa de la frecuencia que te ofrecemos a continuación te ahorra ese tedioso trabajo. Todo lo que tienes que hacer es introducir el texto, sin dejar espacios en blanco entre palabras, y cuando hayas terminado presiona RETURN . Volverá a aparecer la leyenda de introducción del texto y esta vez teclearemos únicamente un asterisco y RETURN . De esta forma, el ordenador presenta una lista de las letras del alfabeto y la cantidad de veces que ha sido utilizada cada una. Como observación, cabe decir que cuando escribimos el texto hay unos instantes de aparente inactividad por parte del ordenador, ocupado en la tarea de procesamiento del mensaje.

A continuación teclea el programa y observa cómo funciona todo ello en la práctica.

```
30 CLS:COLOR 15,13:KEY OFF
   :WIDTH 40
40 PRINT"FRECUENCIA DE
   LOS CARACTERES"
45 PRINT:PRINT "Escribe en
   MAYUSCULAS y ..."
50 PRINT"No dejes espacios
   entre palabras"
60 DIM N(28)
70 PRINT:PRINT"Al terminar
   de escribir el texto"
80 FOR T=1 TO 28:N(T)=0
   :NEXT T
85 PRINT "teclea (*) y
   obtendras el resultado"
87 LOCATE 0,20:PRINT
   "PULSA ESPACIO"
88 R$=INKEY$:IF R$=""
   THEN 88
90 CLS:INPUT"Escribe el
   texto";A$
100 IF A$="*" THEN 180
110 CLS
120 FOR I=1 TO LEN(A$)
130 FOR J=1 TO 26
140 IF J=ASC(MID$(A$,I,1))
    -64 THEN N(J)=N(J)+1
150 NEXT J
160 NEXT I
170 GOTO 90
180 CLS:PRINT"letra frec
    letra frec":PRINT"-----
    ----   ----- ----"
```



```
    :PRINT
200 FOR I=1 TO 13
210 PRINT CHR$(64+I);TAB(5)
    N(I);TAB(13) CHR$(77+I)
    ;TAB(18) N(13+I)
220 NEXT I
```

En cuanto a su funcionamiento, el programa se limita a ejecutar un procedimiento de conteo. En la primera fase del mismo asigna el valor 0 a cada una de las 28 variables indexadas. En dichas variables se almacenan los resultados parciales y totales de las 26 letras. Las dos últimas se incluyen para el caso de que se quiera modificar el programa para obtener porcentajes o cualquier otro dato estadístico aislado.

CODIGO DE MULTIPLICACION

Durante la guerra civil nortamericana se utilizó el código de la empalizada para enviar mensajes secretos. Su funcionamiento es más o menos el siguiente. Supongamos que quieres enviar este mensaje: CONSULTA VENTA URGENTE, que podría tener un interés comercial. En una hoja de papel cuadriculado escribe la primera letra del mensaje en la línea superior, la segunda letra en la segunda línea, la tercera en la primera línea y así, sucesivamente con las demas, hasta obtener:

CNUTVNARET

OSLAETUGNE

que, como verás, recuerda a los postes ferroviarios colocados al tresbolillo. Este mensaje se puede codificar como CNUTVNARET OSLAETUGNE o también puede trocearse en longitudes de palabra más reales, así:

CNUTV NARETOS LAETUGNE

El código de la empalizada es realmente un caso especial de los modernos códigos multiplicativos. Ignorando los espacios, el texto original del mensaje en claro contiene 20 caracteres. Estos pueden ser almacenados en cadenas de 2 ×10, 10 × 2, 5 × 4 ó 4 × 5 elementos, tal como hemos visto.

Todo el que intente romper el códi-



go sin un conocimiento previo de la clave de multiplicación tendrá, sin duda, gran cantidad de trabajo ante sí. Lo que ocurrirá es que iremos llenando la página con el mensaje en sentido vertical, hasta llenar la primera cadena. Seguidamente se comienza con el principio de la segunda y así sucesivamente. Una vez todas llenas podemos descifrar el mensaje. Es por esto que resulta una buena idea el hecho de intercalar algunas letras falsas en el mensaje para dificultar el trabajo de los intrusos.

El programa utiliza también la función de troceo y manipulación de cadenas (entre las líneas 140 a 210). El código de multiplicación simplemente lee el mensaje de una cadena en una dirección y lo visualiza en la otra. Sencillo, pero efectivo.

```
30 CLS:KEY OFF:PRINT
   "CODIGO DE
   MULTIPLICACION"
40 PRINT:PRINT"Escribe
   en MAYUSCULAS ..."
50 PRINT:PRINT
60 INPUT "Escribe el texto
   : ";M$
70 INPUT"Codificar (C) o
   decodificar (D)";E$
80 INPUT "Numero de filas"
   ;N
90 CLS
100 IF E$="C" THEN X=N
110 IF E$="D" THEN X=
    INT(LEN(M$)/N)+1+
    (LEN(M$)MOD N=0)
120 IF E$<> "D" AND E$<>
    "C" THEN GOTO 60
130 DIM D$(X+1)
140 FOR I=1 TO X
150 D$(I)=""
```




```
160 FOR J=1 TO LEN(M$)
    STEP X
170 B$=MID$(M$,J+I-1,1)
175 IF J+I-1 > LEN(M$)
    THEN B$=" "
180 D$(I)=D$(I)+B$
190 NEXT J
200 PRINT D$(I);
210 NEXT I
```

LIBRO DE CLAVES

Hasta ahora solamente hemos considerado los métodos de cifrado. Las claves, propiamente dicho, son palabras o frases completas cifradas por otras palabras o números. Son tradicionalmente empleadas por grandes empresas que operan partiendo de premisas fijas. Las embajadas, los barcos y los negocios caen en esta categoría. Operando desde ubicaciones fijas resulta preferible, porque se precisan uno o más gruesos diccionarios para traducir el texto en bruto.

El programa del libro de claves dispone de un pequeño diccionario de 20



palabras. Lo primero que hace el programa es leer (READ) los datos incorporados en las sentencias DATA, que son las palabras y el número del código que le hacemos corresponder a cada una. Simultáneamente, las va depositando en la cadena bidimensional AS(I,J).

La siguiente sección (líneas 120 a 170) toma una palabra del mensaje y proporciona el número asociado como respuesta. Si, por ejemplo, AS(1,6) = SALIDA obtendremos en la pantalla el contenido del elemento AS(2,6) =«68677».

Un mensaje más largo, que podría ser: SALIDA DESDE NUEVA YORK A MEDIANOCHE. LLEGADA A PARIS DOMINGO MEDIANOCHE, será codificado como: 68677 54982 10996 26569 68719 10996 90075 48553 26569.

Por otro lado, el mensaje en clave: 74891 22317 10996 23874 se traduce como: ENVIA DINERO A ROMA.

Con este sencillo programa será más rápido codificar y decodificar que empleando el método manual, pués cuando las palabras deben ser elegidas entre cientos o miles el ahorro de esfuerzo es evidente.

```
30 CLS:WIDTH40:KEY OFF
40 PRINT"LIBRO DE CODIGOS"
50 DIM A$(2,20)
60 FOR I=1 TO 2
70 FOR J=1 TO 20
80 READ A$(I,J)
90 NEXT J,I
100 PRINT:PRINT"Codificar
    (0) o decodificar (1)"
    :INPUT X
110 PRINT:PRINT"Escribe en
    MAYUSCULAS"
115 PRINT"Utiliza * para
    terminar":PRINT
120 INPUT"Escribe la palabra"
    ;M$
130 IF M$="*" THEN 280
140 FOR T=1 TO 20
150 IF M$=A$(1+X,T) THEN
    PRINT A$(2-X,T)
160 NEXT T
170 GOTO 120
180 DATA NUEVA YORK,
    LONDRES, PARIS,ROMA
190 DATA LLEGADA,SALIDA,
    DESDE,VA A,ESCAPA,SABADO
200 DATA DOMINGO,TARDE,
    AMANECER,MEDIANOCHE
210 DATA NOCHE,A,EN EL,
    ENVIA
220 DATA DINERO,COMIDA
230 DATA 54982,73581,90075
    ,23874
240 DATA 68719,68677,10327
    ,40476
250 DATA 27921,48553,11072
    ,70355
260 DATA 26569,74832,10996
    ,12128
270 DATA 69783,74891,22317
    ,98724
280 END
```

De todas formas es posible aumentar el número de palabras a más de 20, basta con variar las cifras que van entre paréntesis detrás de DIM AS, aumentar la cantidad de palabras y códigos en las sentencias DATA y alterar los bucles FOR I y FOR J.

EL ZOCO DE INPUT

Todo se compra y se vende. Los antiguos zocos fueron lugares destinados a todo tipo de transacciones. INPUT también tiene el suyo. Vuestras operaciones de compra, cambio o venta serán publicadas en esta sección, pero dos son las limitaciones que imponemos:

a) La propuesta tendrá que ver con la microinformática.

b) Nos reservamos el derecho de no publicar aquellos insertos de los que se sospeche un trasfondo lucrativo.

Ahora un ruego. Tratar de resumir al máximo el texto; escribir casi como un telegrama siendo claros y concisos.

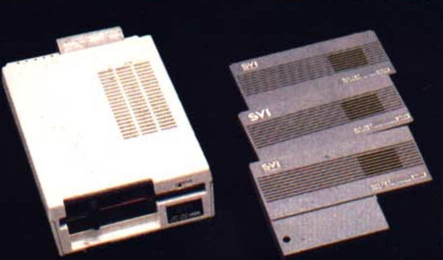


Envía tu mensaje a:

INPUT MSX ZOCO
c/. Alberto Alcocer, 46
28016 MADRID



728







- Capacidad de Memoria de 80 K RAM y 32 K ROM, con el intérprete de BASIC (MICROSOFT) incorporado.
- Teclado de diseño ergonómico, con teclas numéricas independientes.
- Gráficos de alta resolución: red de 296 por 152 puntos, con 16 colores disponibles.
- Conexiones para TV, monitor, unidad de disco de 320 K. (incluido sistema operativo CP/M), impresora, cassettes y cartuchos MSX.
- Conector para dos Joysticks.
- Ordenador MSX.

Aquí y ahora, con toda su potencia

C&M

SVI 738 X'press





- Sistemas operativos: CP/M, MSX-BASIC y MSX-DOS.
- Con una memoria de 80 K RAM.
- Unidad de disco de 3' 5'' integrada en la consola del teclado.
- Dos puertas de conexión: RS232-C y Paralelo Centronics.
- Salidas directas a televisor y monitor.
- Admite directamente la conexión de una segunda unidad de disco, sin necesidad de interface o cartucho de ampliación de memoria.
- Posibilidad de operar en 40 ó 80 columnas sin necesidad de cartucho.
- Transporte cómodo: asa incorporada y maletín de transporte.

SVI

SPECTRAVIDEO

Desde ahora SVI-Spectravídeo está aquí.

Con toda la potencia de su nombre y su organización mundial, con toda su tecnología de futuro.

SVI-Spectravídeo marca el comienzo de una nueva era.

Muchas cosas se van a quedar en el pasado.

Y, probablemente, nada será igual a partir de ahora.

Con SVI-Spectravídeo, la vida diaria puede hacerse más sencilla, con más posibilidades, más divertida y excitante, tanto en casa como en el trabajo.

Porque, desde ahora, y para el futuro, puedes contar con SVI-Spectravídeo, España.

Aquí, con toda su potencia.

Y, siempre, muy cerca de ti: en 1.500 puntos de venta.

SVI-Spectravídeo, España, significa la más avanzada tecnología, mantenida por el servicio técnico más exigente y eficaz, en:

Ordenadores. Periféricos y accesorios. Joysticks. Juguetes electrónicos. Sonido Hi-Fi.



EXIGELE A TU BASIC

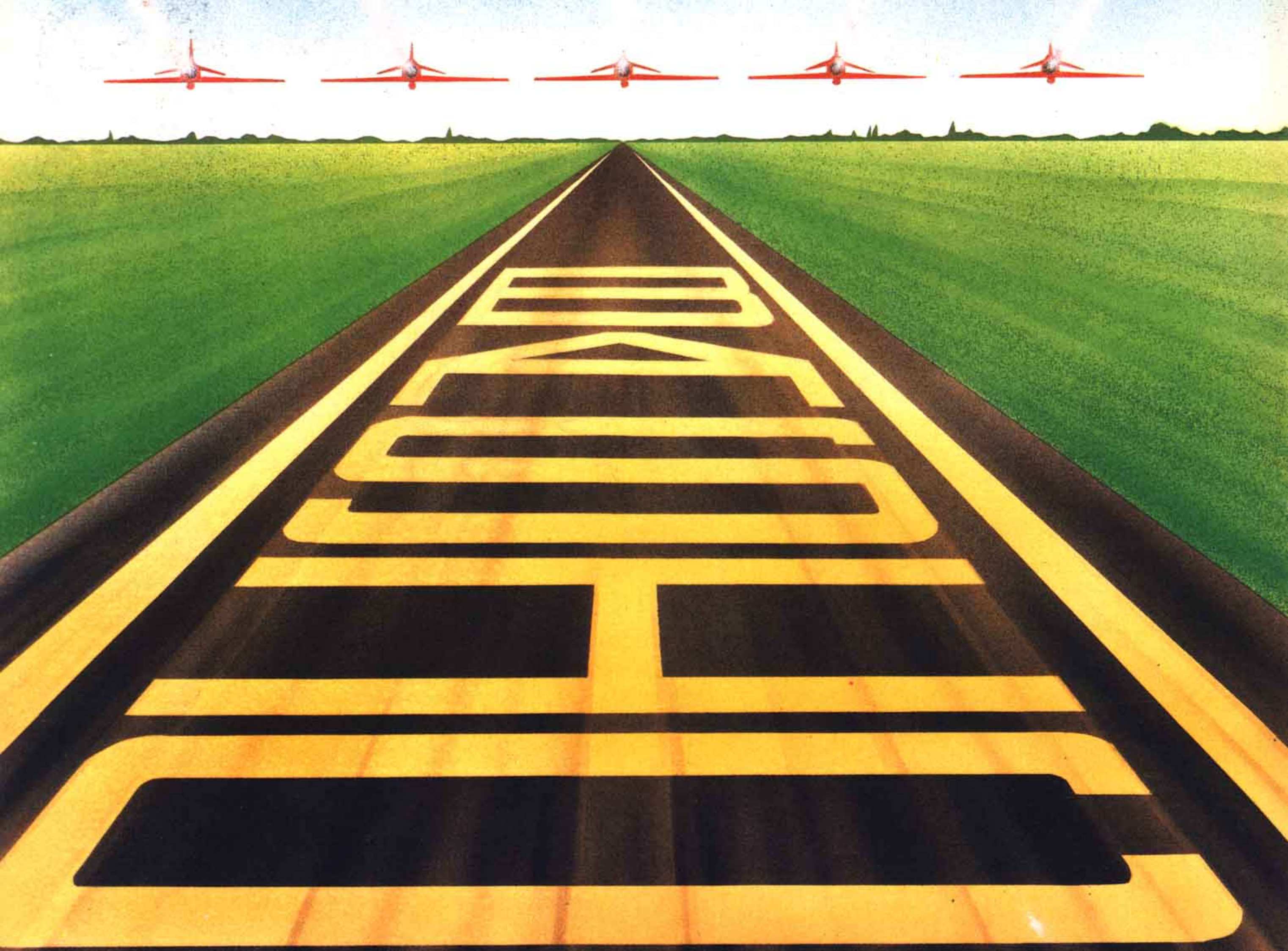
- HAZ QUE TUS PROGRAMAS EN BASIC CORRAN MAS
- TIEMPO DE EJECUCCION DE LAS SENTENCIAS
- ATENCION A LA ESTRUCTURA

Ponte el casco, la visera y las ropas de viaje. Limpia tus bujías y asegúrate de que tu máquina corre con suavidad. Está a punto de bajarse la bandera y ya es tiempo de acelerar tu BASIC.

Al principio, todos los ordenadores parecían rápidos como un rayo, realizando «en un abrir y cerrar de ojos» tareas que a cualquier persona le requerirían mucho más tiempo. Pero prueba a escribir un juego de acción, o hacer que la máquina ejecute una larga serie de cálculos, o un trabajo complejo de clasificación, con el BASIC y te encontrarás con una situación muy distinta. Verás que realmente la máquina tarda en terminar la tarea que se le ha asignado. Puede que muy pronto empieces a quejarte de lo lenta que es.

Los programas que se ejecutan con más rapidez son los que están escritos en código máquina o lenguaje ensamblador, si bien la mayor parte de la gente encuentra más fácil programar en BASIC. Desgraciadamente, un programa en BASIC nunca puede esperar ni siquiera acercarse a la velocidad de un programa escrito en código máquina, debido a que el ordenador tiene que gastar tiempo traduciendo las instrucciones del BASIC al código máquina. Para ello lleva incorporado un programa especial que se ocupa de este trabajo: el **intérprete**.

Si tú no quieres escribir programas en código máquina, pero sí apurar al máximo la velocidad de tu máquina, puedes ensayar varios trucos. En primer lugar, procura estructurar adecuadamente tus programas. En segundo lugar, tienes que intentar conseguir que cada línea individual de programa opere a la velocidad óptima. Elige





ML-FX1/2 ☐

El MSX profesional
80 Kb RAM.
Teclado Numérico.
ML-FX2 Programa MAP (B. Datos/ P. Textos / H. Cálculo Graficos/ Comunicaciones.

ML-30 FD ☐

La Máxima capacidad en disco.
1 Mb. (720 Kb. Formateado)
8 Formatos diferentes
Chasis previsto para 2 unidades.

ML-10 DR ☐

Cassette especial para ordenador.
Admite 1200/2400 baud.
Cuentavueltas. Señal de monitor.
Alimentación a red o baterias.

ML-10 MA ☐

Ratón para diseño gráfico.
Programa CHEESE de diseño.
24 Funciones gráficas.

APLICACIONES ☐

Un Software profesional para un ordenador profesional. Contabilidad, Control de Stock, Facturación.

CT-1501 E ☐

Monitor/Televisión.
Alta definición.
Conector SCART.
Mando a distancia.

CUPON DE RESPUESTA

Desearía poder tener más información sobre los aparatos marcados ☒ de MITSUBISHI.

Sr.: ____________________

____

Domicilio: ____________________

____

Población: ____________________

MABEL, S.A.
Pº Maragall, 120 - 08027 BARCELONA

pues los componentes del BASIC que el intérprete traduce con más rapidez.

Cada máquina tiene sus propias peculiaridades, y en cierta medida, cada programa tiene sus propias exigencias. En consecuencia no existen reglas generales e infalibles para conseguir el programa perfecto. Sólo se pueden dar ciertas indicaciones, a tí te corresponde escoger los trucos a incorporar a tus programas, ya que el uso de los mismos puede requerir sacrificar otras cosas.

MEDIDA DE TIEMPOS EN PROGRAMAS DE BASIC

Todas las máquinas llevan incorporado un reloj que puede utilizarse para comparar la rapidez de unos programas respecto a otro. Teclea la rutina siguiente, especialmente diseñada para tu máquina, para que puedas ir viendo por tí mismo en los ejemplos posteriores de este artículo las diferencias entre formas alternativas de programar en BASIC:

```
 10 T=TIME
 20 FOR I=1 TO 100:GOSUB
    200:NEXT I
 30 T=TIME-T
 35 CLS
 40 PRINT "Tiempo":PRINT
    "Transcurrido:";(T-15)
    /.05;" Milisegundos"
 50 END
200 REM
500 RETURN
```

La rutina lee en la línea 10 el valor del reloj interno de tu **MSX**. A continuación, en la línea 20, se repite 100 veces el salto a la subrutina que empieza en la línea 200. En este caso la subrutina es sólo una instrucción REM.

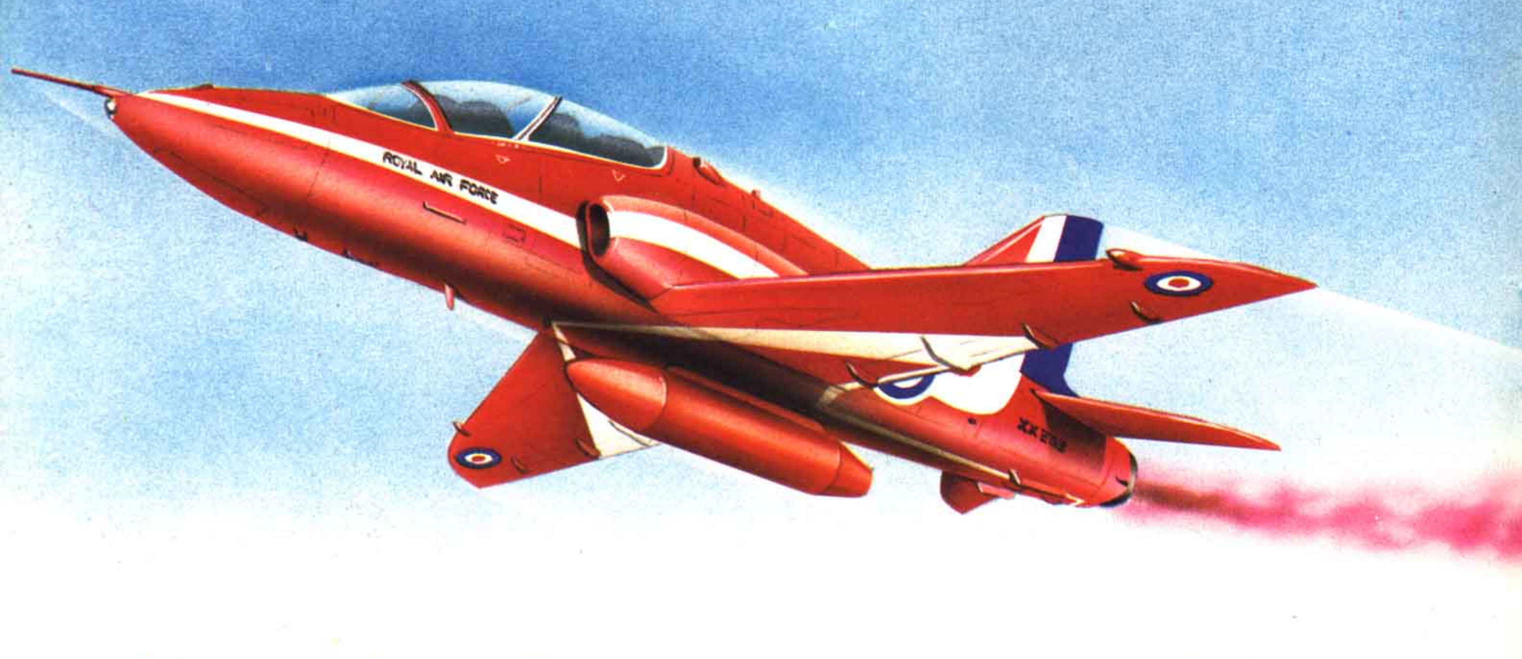
Al terminar, se vuelve a leer el valor del reloj, en la línea 30, y se resta del valor leído al principio. El resultado es el tiempo que ha tardado en ejecutarse la subrutina, tiempo que se imprime en la línea 40. Cuando más adelante cambiemos la instrucción REM por otra instrucción, sabremos cuanto tiempo ha tardado en ejecutarse 100 veces dicha instrucción.

ESTRUCTURA

Aunque ya hemos hablado bastante en INPUT sobre los programas estructurados, merece la pena insistir en aquellos factores que más influyen en la velocidad de ejecución.

Todas las subrutinas que se utilicen con frecuencia, tienes que situarlas cerca de principio del programa. La razón de esto es que el intérprete empieza a buscar el número de la línea que aparece junto al GOSUB desde el principio. Obviamente, si el número de línea es bajo, será encontrada con más rapidez que si es alto. El ahorro de unos milisegundos aquí y allá puede convertirse enseguida en una contribución significativa en cuanto a velocidad.

Los programas mal pensados que utilizan un laberinto indiscriminado de GOTOs —a los que a veces se llama programas de *spaghetti*— no sólo constituyen una seria amenaza para



los que intenten leer el programa, sino que pueden ser un gran obstaculo para la ejecucción rápida. Asi pues, un programa bien planeado es probable que sea más rápido de ejecución que uno que has ido modificando una y otra vez sin un plan previo.

MEMORIA Y VELOCIDAD

En general, los programas cortos es muy probable que se ejecuten también más rápidamente. Sin embargo, las tres condiciones principales que debe cumplir un programa —velocidad, claridad y economía de memoria— suelen estar normalmente en conflicto. Por ejemplo, las líneas de programa con muchas sentencias, sirven para ahorrar memoria y aumentar la velocidad, pero pueden hacer que el listado sea más difícil de seguir y depurar.

Un programa diseñado para tener máxima velocidad de ejecución, es posible que aumente mucho en longitud y en uso de memoria. Recíprocamente, algunos programas que son muy efectivos para ahorrar memoria, lo consiguen a expensas de una reducción de velocidad.

Se puede ahorrar mucha memoria utilizando subrutinas, pero las llamadas a subrutinas consumen tiempo y por lo tanto son indeseables si lo que quieres es una ejecución muy rápida. Como ya has visto, si optas por las subrutinas, tienes que ser muy cuidadoso con la estructura del programa para poder recuperar algo de la velocidad perdida.

Análogamente, A = VAL («100») está muy bien para ahorrar memoria pero resulta desastrosa cuando se compara con A = 100, que consume más memoria, pero está más indicada para aumentar la velocidad.

VARIABLES

Si quieres acelerar tus programas, te resultará muy útil el conocimiento de cómo se almacenan las variables en la memoria. El área de variables se limpia con RUN o CLEAR, y las variables se crean a medida que van apareciendo. Generalmente las nuevas variables se añaden extendiendo hacia arriba el área de variables.

Consideremos una situación en la que se crea una variable que es una cadena de caracteres y a continuación una matriz númerica. Después puedes añadir algo a la cadena de caracteres. Si la cadena ya había sido creada antes en el programa, todas las variables creadas con posterioridad tendrán que desplazarse hacia arriba en la memoria, para acomodarse a las condiciones de la cadena.

Echa un vistazo a este ejemplo:

```
100 T$=""
110 DIM A(1000)
120 T$=T$+"LO QUE SEA"
```

En casi todas las versiones de BASIC se puede ahorrar bastante tiempo permutando las dos primeras líneas de programa, es decir, dimensionando la matriz antes de definir la cadena de caracteres. Tal como está el programa, puede que haya que mover 5000 bytes cada vez que se suma «LO QUE SEA» a T$.

Hay una regla general, por la que si usas variables en vez de números, ahorrarás gran cantidad de tiempo.

En el caso del BASIC **MSX**, parece que ocurre lo contrario.

A = B + B, donde B = 10, tarda en ejecutarse unos milisegundos más que A = 10 + 10.

Estos poco milisegundos pueden no tener importancia en la mayoría de los casos, pero si el programa incluye muchos bucles en los que se repite muchas veces este caso, los milisegundos se suman y al final el tiempo perdido puede ser considerable.



P y R

¿Por qué estas reglas de velocidad se aplican solamente a algunas máquinas?

No hay normas rígidas sobre cómo escribir los programas en BASIC para arrancarle el esfuerzo supremo de velocidad, como tampoco las hay sobre la forma de escribir intérpretes; después de todo son programas de máquinas. Pueden contener compromisos parecidos a los de los programas que tú puedas escribir.

La velocidad de un intérprete dependerá de la forma en que esté escrito y de las prestaciones que el fabricante quiera incorporar en su BASIC.

P y R

El BASIC me sigue pareciendo demasiado lento, pero creo que programar en código máquina es muy tedioso. ¿Qué otra cosa puedo hacer?

Hay una especie de término medio, constituido por los lenguajes compilados en vez de interpretados. Lo que más tiempo requiere es ir interpretando el BASIC mientras el programa está corriendo.

El programa compilado utiliza en cambio otro tipo de programa —parecido a los familiares ensambladores de la programación en código máquina— que convierte el programa escrito en un lenguaje de alto nivel en un programa en código máquina, antes de empezar la ejecución. Esto significa que después de la compilación, se ejecuta un programa en código máquina en vez de un programa de alto nivel.

FUNCIONES MATEMATICAS

Utiliza las rutinas de medida de tiempos para comparar estas dos maneras de calcular la misma cosa:

```
200 C=4*4*4*4
```

o bien

```
200 C=4^4
```

Como habrás podido comprobar, en tu **MSX**, al contrario que en muchos otros ordenadores, la función de potencialización incorporada en el BASIC, resulta más rápida que si utilizas la función producto. Y además de ser más rápida, ahorra memoria. Fíjate que en el primer caso hay que almacenar cuatro números y en el segundo sólo dos.

Se ha sugerido a veces que líneas como la siguiente:

```
210 IF X<Y THEN Y=Y+1
220 IF X>Y THEN Y=Y-1
```

pueden ganar mucho en velocidad sustituyéndolos por:

```
210 Y=Y+(X<Y)-(X>Y)
```

Efectivamente, si pruebas con la rutina de medida de tiempos, verás que el ejemplo de una sola línea es algo más rápido. La diferencia es muy pequeña, por lo que para comprobarla tendrás que aumentar el número de repeticiones cambiando el valor 100 de la línea 20, en la rutina de medida.

Como además una sola línea es más elegante y ahorra espacio de memoria, no hay duda de que conviene utilizarla siempre que sea posible.

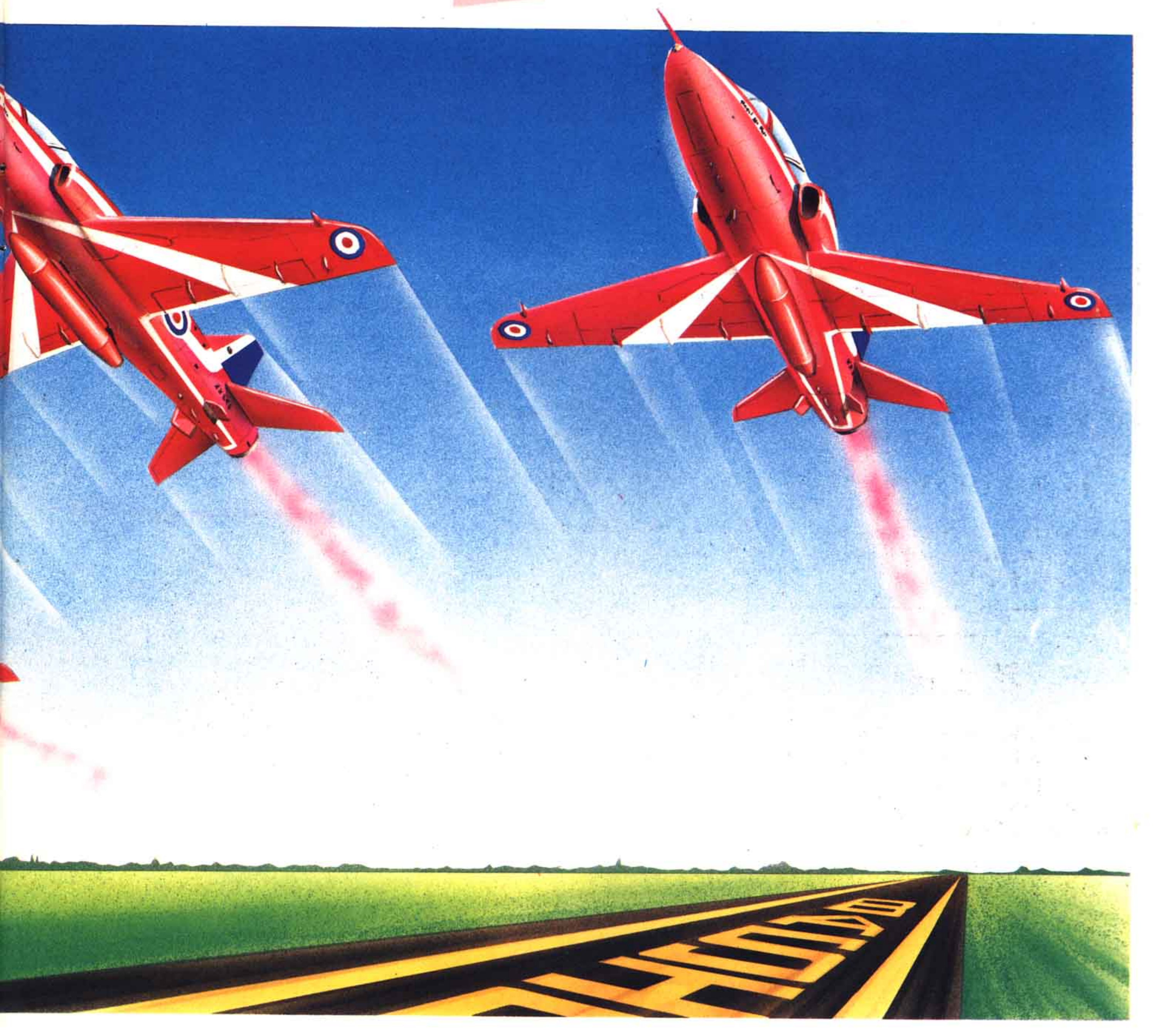


MULTIPLICACION Y DIVISION

Las expresiones C = D*0.5 y C = D/2 realizan exactamente el mismo cálculo, pero te encontrarás que la multiplicación es algo más rápida.

Prueba las siguientes sugerencias para ver cuál es más rápida y anota los resultados por si te hacen falta en el futuro:

```
200 C=10+10
200 C=D+D (Donde D=10)
200 C=10*10
200 C=10/10
200 C=SIN (10)
200 C=COS (10)
200 C=TAN (10)
200 C=VAL ("10")
200 C=10
200 C=D   (Donde D=10)
200 PRINT "TEST"
200 PRINT A$
    (Donde A$="TEST")
200 PRINT 10+1000+500
200 PRINT A+B+C
    (Donde A=10, B=1000,
    y C=500)
```



CLASIFICACION Y BUSQUEDA

En un próximo artículo de INPUT nos ocuparemos en profundidad del tema de la clasificación. La ordenación de **Shell-Metzner**, que veremos en dicho artículo, sería la más adecuada para utilizar en cualquier aplicación que implicara la ordenación de, por ejemplo, más de cien elementos. Tiene la particularidad de que cuanto más datos tienes que manejar, más rápida es clasificando. Es desde luego un método mucho más rápido que el conocido método de la burbuja.

Muchas veces se considera el problema de la búsqueda junto con el de la ordenación. En este caso, en vez de poner en orden una colección de datos, lo que se pretende es recuperar un elemento en particular (o un conjunto de elementos relacionados) con la mayor rapidez posible. Suponte que tienes en tu ordenador una lista de números de teléfono, y que necesitas el número de Alberto Gómez e Hijos, Papeles Pintados. Naturalmente lo que quieres es que la máquina tarde menos en encontrar el número que si tienes que buscarlo en la guía de teléfonos. Por eso interesan rutinas lo más rápidas posible.

La búsqueda en serie, hace lo mismo que una persona que recorre una lista en un papel, nombre a nombre.

```
10 REM**BUSQUEDA SERIE
20 DIM B$(10)
30 CLS:FOR I=1 TO 10:READ
   B$(I):PRINTI,B$(I):NEXT I
35 FOR J=1 TO 1500:NEXT J
```

REGLAS PARA AUMENTAR LA VELOCIDAD DEL BASIC

- Empieza las matrices a partir del subíndice 0 y no a partir del 1.
- En los bucles FOR ... NEXT, no pongas variables después del NEXT.
- Define las matrices al principio del programa.
- Siempre que sea posible, utiliza variables enteras en vez de con valores decimales.
- Utiliza nombres cortos para las variables. Siempre que sea posible, una sola letra.
- Coloca todas las subrutinas de uso frecuente al principio del programa.
- Con programas de cálculo ten cuidado si defines funciones como por ejemplo:

```
10 DEF FN A(X)=X-X/360
20 B=FN A(100)
```

A veces resulta más rápido utilizar esta otra construcción:

```
10 X=100
20 B=X-X/360
```

- Usa números pequeños para las líneas. Empieza a programar a partir de la línea 10 y no a partir de la 1000.
- Emplea las cortas rutinas en código máquina a las que se llama desde el BASIC para el *scrolling*, etc. Conviértete en un ávido coleccionista de las rutinas que se publican en código máquina.
- Planifica tu programa sobre el papel antes de empezar a teclear. De esta forma evitarás la pérdida de tiempo que suponen los excesivos saltos arriba y abajo, que son síntomas de un programa mal planeado.
- Recuerda que las subrutinas ahorran memoria pero gastan tiempo.
- Elimina todos los denominadores comunes superfluos. Sustituye

```
10 X=Y/100+Z/100 por
10 X=(Y+Z)/100
```

- Evita la sentencia GO TO siempre que sea posible.
- Siempre que puedas, escribe varias sentencias en una línea.
- Elimina del programa todos los espacios innecesarios, las líneas en blanco y las sentencias REM.
- En las sentencias IF, coloca primero las condiciones con más probabilidades de ser falsas.
- Reutiliza los nombres de variable y las variables de bucle, en vez de definir otros nuevos.

```
40 CLS:PRINT"ESCRIBE LA
   PALABRA ":INPUT "EN
   MAYUSCULAS ";A$
50 FOR X=1 TO 10
60 IF B$(X)=A$ THEN PRINT
   :PRINT"LA ENCONTRE":GOTO
   75
70 NEXT X
75 END
80 DATA "ANTILOPE","BURRO",
   "CABRA","DELFIN",
   "ELEFANTE"
85 DATA "GORRION","IGUANA",
   "KOALA","PERRO","GATO"
```

Aunque la búsqueda serial es una rutina muy conocida, no es muy rápida. La búsqueda binaria o dicotómica no es mucho más difícil de programar, pero es de ejecución mucho más rápida, si bien la diferencia se advierte mejor cuando las listas son largas.

La mayor velocidad de la búsqueda binaria se debe a que, a diferencia de la búsqueda en serie, no tiene que examinar todos los elementos de la lista. Previamente los datos tienen que estar ordenados por orden alfabético o numérico, y el ordenador mira primero al elemento que hay en el centro de la lista. Desde este punto se mueve hacia arriba o hacia abajo, reduciendo a la mitad cada vez la lista restante, ya que va comparando cada elemento que se encuentra con el que va buscando. Inicialmente no busca un coincidencia perfecta, como en la búsqueda serie, sino que simplemente observa si la primera letra es más alta o más baja que la del elemento deseado.

```
10 CLS
20 T%=10:B%=1
30 DIM N$(10)
40 FOR C=1 TO 10
50 READ N$(C):PRINT C,N$(C)
60 NEXT C
65 FOR C=1 TO 1000:NEXT C
70 CLS:PRINT"ESCRIBE TU
   ANIMAL":INPUT"EN
   MAYUSCULAS";A$
80 TIME=0
90 IF TIME<50 THEN 90
95 CLS
100 IF N$(T%)=A$ THEN PRINT
    N$(T%),T%:GOTO 200
110 IF N$(B%)=A$ THEN PRINT
    N$(B%),B%:GOTO 200
120 P%=(T%+B%)/2
130 IF N$(P%)=A$ THEN PRINT
    N$(P%),P%:GOTO 200
140 IF N$(P%)>A$ THEN T%=P%
150 IF N$(P%)<A$ THEN B%=P%
160 IF T%-B%=1 THEN PRINT
    "NO LO ENCUENTRO":GOTO 20
    0
170 GOTO 100
200 LOCATE 0,15:PRINT"PULSA
    UNA TECLA":G$=INKEY$:IF
     G$="" THEN 200
210 RUN
500 DATA "ANTILOPE","BURRO",
    "CABRA","DELFIN",
     "ELEFANTE"
510 DATA "GORRION","IGUANA",
    "KOALA","PERRO","GATO"
```

Para acelerar los programas de BASIC no hay una respuesta única. Los mejores resultados se obtienen con una atención cuidadosa a muchos detalles aparentemente insignificantes y con mucha experimentación. Como cada programa es un caso especial, nunca puede haber reglas infalibles sobre la forma en que hay que escribir programas. Siempre hay cosas nuevas por descubrir y en parte a ello se debe la fascinación de la programación. Es muy probable que los mayores aumentos de la velocidad de ejecución de tus programas se deban a tus propios descubrimientos.

RUTINAS DE TANTEO Y TIEMPO

No hay nada como saber que sólo dispones de dos segundos para regresar a tu base, o que te faltan diez puntos para obtener la máxima puntuación en tu juego de batallas. Todos los juegos de marcianitos llevan algún tipo de cuenta de tanteo y tiempo, para darle más emoción. Con algunos programas sencillos, tú puedes hacer lo mismo.

Casi todos los juegos de ordenador necesitan algún tipo de puntuación o temporización, o incluso ambas cosas. Sin ellos no puedes juzgar lo bien que se te da el juego, o si vas mejorando algo, y no suele tener mucho interés jugarlo con tus amigos.

Podrías tener a alguien sentado tras de ti que fuera contando los impactos que consigues sobre tu feroz enemigo, pero esto no tiene mucho sentido cuando puedes programar a tu ordenador para que los cuente él. Con unas cuantas líneas de programa más, la máquina recordará también las puntuaciones.

Por la misma razón no hay necesidad de recurrir a un cronógrafo para la medida del tiempo. Todas las máquinas llevan un reloj incorporado, y puedes servirte de él de muchas formas para mejorar tus juegos.

CAMPO DE MINAS

Para que veas la manera de incorporar en la práctica las rutinas de puntuación y temporización, aquí tienes un juego en el que se van añadiendo las rutinas progresivamente. Cada rutina es muy sencilla y se puede añadir también a otros juegos.

El juego se llama Campo de Minas, y en él tú vas conduciendo un carro de combate, cuya misión es rescatar a unos paracaidistas que se han arrojado temerariamente sobre un campo minado. Cada vez que el tanque se



mueve, corre el riesgo de hacer detonar una mina plantada aleatoriamente por el ordenador. Como en un campo minado de verdad, las minas son invisibles, por lo que tendrás que moverte con precaución.

El tanque (desafortunadamente es sólo un signo #, hasta que aprendas la manera de combinar gráficos y movimiento en un programa BASIC) se controla utilizando las teclas de movimiento de cursor. Para ello se hace uso de la funcion STICK(0) del BASIC **MSX**, que se encarga de leer las teclas de cursor y de proporcionar uno u otro valor según la tecla que se haya pulsado.

De hecho, el nucleo del programa está constituido por la rutina de «moverse por la pantalla» que ya conoces.

Cuando teclees esta sección del juego y la ejecutes (con RUN), verás que todavía no está completa: después de que hayas rescatado al paracaidista, no sucede nada, excepto que el tanque continúa vagando sin rumbo por la pantalla. El programa ha dedetenerse pulsando la tecla CTRL/STOP o tendrás que esperar hasta que el tanque tropiece con una mina escondida. Pero no te alarmes, todo esto mejorará en cuanto le pongas las rutinas de puntuación y de tiempo que siguen.

```
5 TX=16:TY=5
10 WIDTH 40
12 A=RND(-TIME)
15 COLOR 15,4
80 CLS:LOCATE 0,12:PRINT "---
   --------------------------
   -----------"
90 PX=INT(RND(1)*30)+1
100 PY=INT(RND(1)*10)
110 IF PX=TX AND PY=TY THEN
    GOTO 90
120 LOCATE PX,PY:PRINT"O"
    :LOCATE TX,TY:PRINT"#"
130 AX=TX:AY=TY
140 A=STICK(0)
145 IF A=1 THEN TY=TY-1
150 IF A=5 THEN TY=TY+1
160 IF A=7 THEN TX=TX-1
170 IF A=3 THEN TX=TX+1
190 IF TY<0 OR TY>11 THEN
    TY=AY
200 IF TX<0 OR TX>39 THEN
    TX=AX
230 LOCATE AX,AY:PRINT" "
240 LOCATE TX,TY:PRINT"#"
250 MX=INT(RND(1)*30)+1
260 MY=INT(RND(1)*10)
270 IF MX=TX AND MY=TY THEN
    LOCATE MX,MY:PRINT" "
    :LOCATE 0,14:PRINT
    "BOOM!!-TE ALCANZO UNA
    MINA":STOP
310 GOTO 130
```

El programa comienza dividiendo la pantalla en dos mitades, mediante una línea de trazos que se dibuja desde la línea 80 del programa. Previamente, entre las líneas 5 y 15 se ha definido el color y la anchura de la pantalla así como la posición inicial del tanque en las variables TX y TY.

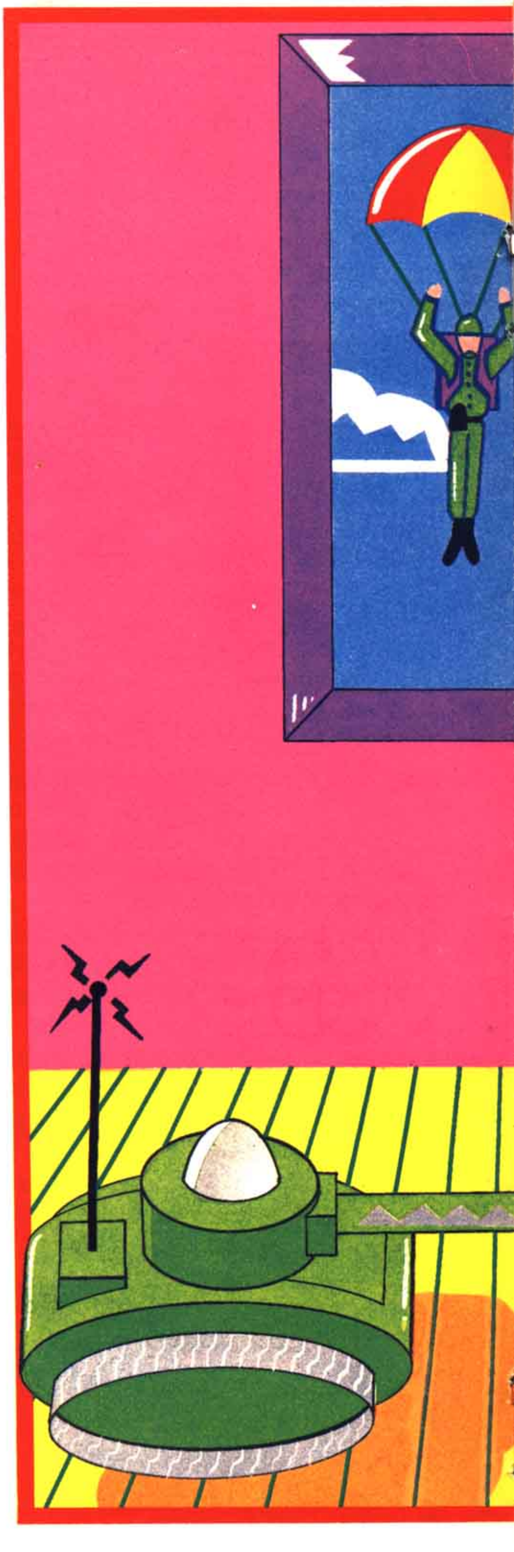
En las líneas 90 y 100 se eligen aleatoriamente las coordenadas del punto de caída del paracaidista. Esta posición se compara con la que ocupa el tanque en la línea 110. Si son iguales, se elige una nueva posición para el paracaidista. Tanto éste como el tanque se dibujan en la pantalla al ejecutarse la línea 120.

Entre las líneas 140 y 170 nos encontramos con la rutina de lectura de las teclas de cursor.

Para evitar que el tanque pueda salirse de los límites de la pantalla se utilizan las líneas 190 y 200.

En las líneas 250 y 260 se elige aleatoriamente una posición para la mina. A continuación, en la línea siguiente, se comparan las posiciones de la mina y del tanque. Si ambas coinciden se produce la explosión y aparece en la pantalla la palabra BOOM.

La línea 310 devuelve el programa al principio del bucle de lectura de teclado.



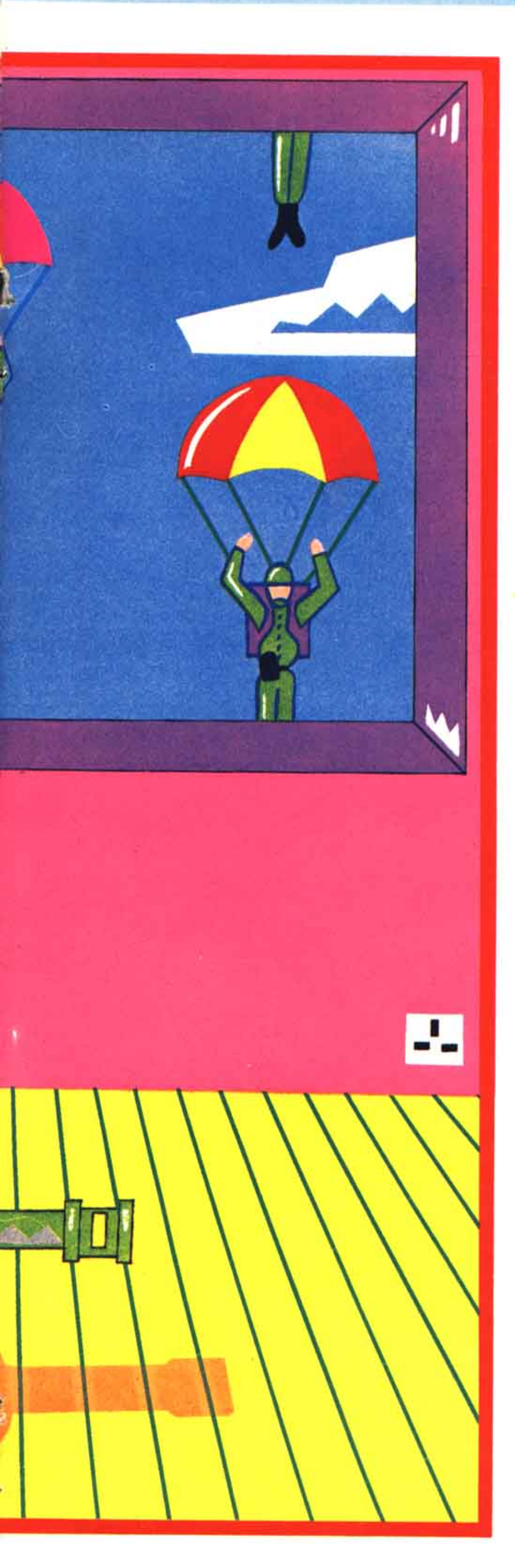
PUNTUACION

En los juegos de extraterrestres, la puntuación aumenta normalmente cuando la posición ocupada por dos objetos en la pantalla es la misma.

Los objetos pueden ser un misil y un blanco, un comecocos y una píldo ra alimenticia, un tanque y un paracaidista, o lo que el juego requiera.

Añade, pues, estas líneas a tu programa para ver cómo trabaja en la práctica el mecanismo de tanteo.

```
40 P=0
280 IF PX=TX AND PY=TY THEN
```



```
    P=P+1:GOTO80
330 LOCATE 0,16:PRINT P;
    "Paracaidistas recogidos"
```

Cambia el STOP de la línea 270 por un GOTO 330. La línea 270 se convierte ahora en:

```
270 IF MX=TX AND MY=TY THEN
    LOCATE MX,MY:PRINT" "
    :LOCATE 0,14:PRINT
    "BOOM!!-TE ALCANZO UNA
    MINA":GOTO330
```

La línea 280 es importante. En ella se comprueba que el tanque y el paracaidista ocupan la misma posición en la pantalla. Si ocurre esto, la puntuación aumenta en 1.

En la línea 40 se pone a cero el tanteo antes de que empiece el juego y la línea 330 visualiza la puntuación. Cambiando el STOP de la línea 270 se consigue que el ordenador presente la puntuación después de haber tocado una mina.

El jugador se enfrenta ahora con una sucesión de paracaidistas a los que rescatar. Cada vez que es rescatado uno, cae otro del cielo. El juego se detiene cuando se produce la explosión de una mina (porque el tanque ocupa la misma posición en la pantalla).

PUNTUACION MAXIMA

No es difícil añadirle a tu juego una opción de puntuación máxima. No tienes más que introducir una variable asociada a esta puntuación —por ejemplo, PM— y algún método para actualizarla cuando sea superada dicha opción, además hay que poner una rutina de presentación.

Aquí tienes las líneas que debes añadir para tener una opción de puntuación máxima.

```
30 PM=0
350 IF P>PM THEN PM=P
370 LOCATE0,17:PRINT"Mayor
    puntuacion";PM
```

En primer lugar debes poner la puntuación máxima en su menor valor posible, por lo que la línea 30 pone PM a cero. Después de que el juego se ha detenido, la línea 350 compara la última puntuación (P) con la puntuación máxima (PM). Si la puntuación obtenida es mayor que la puntuación máxima se actualiza PM, haciéndola igual a P. Finalmente, la línea 370 visualiza en la pantalla el valor.

Es probable que estas líneas te parezcan suficientes para dotar el juego de un tanteo. Por desgracia, esto no es cierto. Cada vez que hagas ejecutar el programa (con RUN), el ordenador olvida automáticamente el valor de PM, y los valores de las otras variables. Para mantener el valor de PM tienes que añadir las líneas de «¿Otra vez?» que se describieron en un capítulo anterior.

```
390 FOR F=1 TO 1000:NEXT F
410 LOCATE 0,19:PRINT"Otra
    vez ? (S/N)"
420 A$=INKEY$:IF A$="" THEN
    420
430 IF A$="s" THEN GOTO 40
440 IF A$="n" THEN CLS:END
450 GOTO 420
```

He aquí lo que hacen estas nuevas líneas:

Hay un corto retardo introducido por la sentencia FOR ... NEXT en la línea 390. El mensaje «Otra vez» (S/N) se presenta en la línea 410.

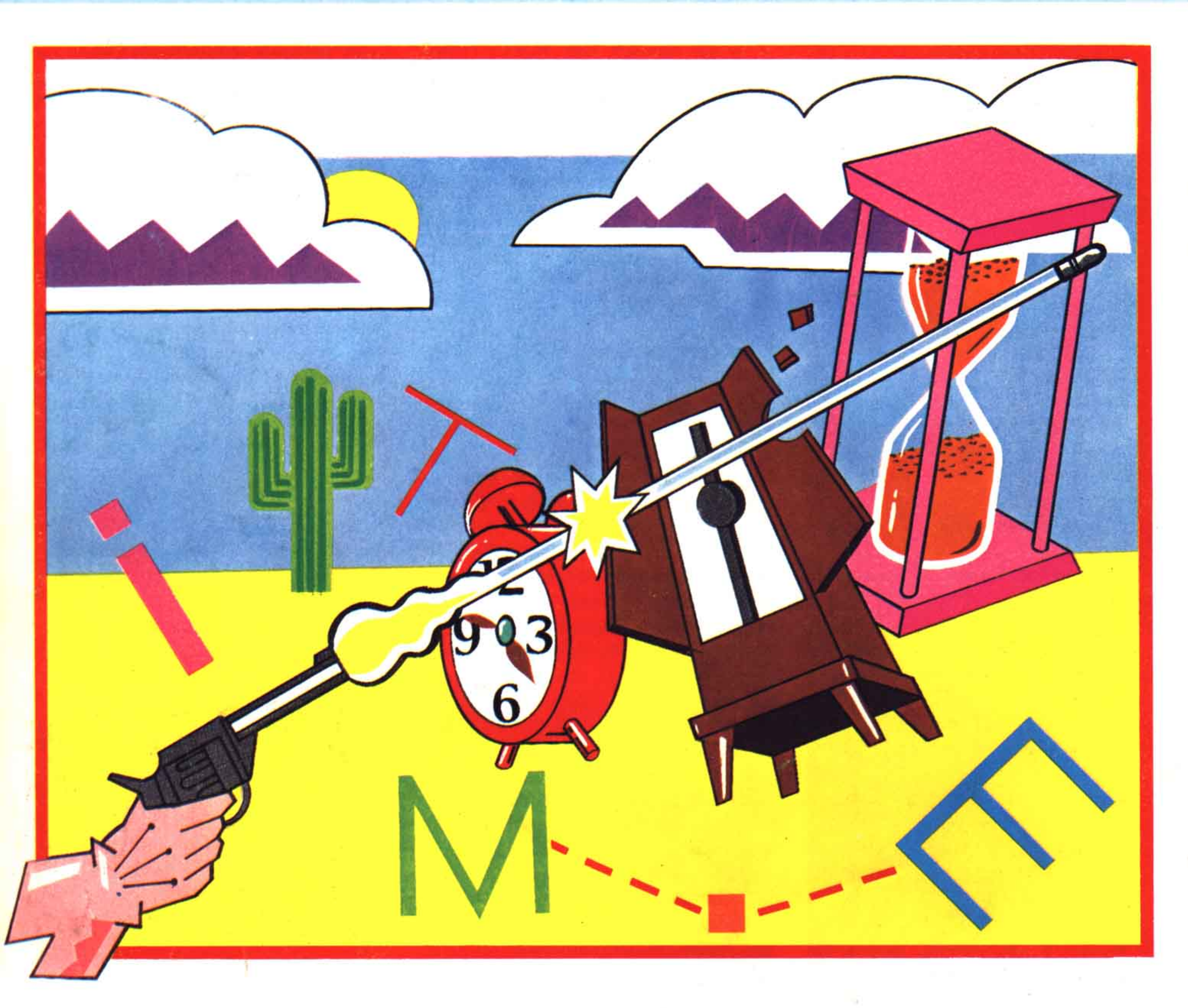
La rutina «Otra vez» está en las líneas 410 a 450. La línea 430 hace que el programa reeomience en la línea 40 si se pulsa S, y la línea 440 detiene el programa si se pulsa N. La línea 450 sirve para asegurarse de que cualquier otra tecla será ignorada.

Como no hay necesidad de pulsar RUN cada vez que desees jugar de nuevo, el valor de PM será preservado, aunque al volver a cargar el programa (con LOAD) se perderá el valor de PM, incluso si encuentras alguna forma de arrancar el programa sin pulsar RUN.

MEDIDA DE TIEMPOS

Tal como está el juego, depende demasiado de la suerte, simplemente, el jugador sigue adelante hasta que pisa una mina escondida.

Se puede introducir en este tipo de juegos un elemento de habilidad, para convertirlo en una carrera contra el reloj. Con los siguientes añadidos puedes cronometrar cuánto tardas en rescatar a diez paracaidistas.




```
75 TIME=0
290 IF P<10 AND TX=PX AND
    TY=PY THEN GOTO 80
300 IF P=10 THEN GOTO 320
320 T=TIME/50
340 IF P=10 THEN LOCATE 0,20
    :PRINT T;"Segundos"
```

El reloj interno de la máquina está corriendo todo el tiempo que el ordenador permanece encendido. Para arrancar el contador de tiempos, tienes que poner a cero la lectura de dicho reloj. Esto se hace en la línea 75. Simplemente teclea TIME=0

El reloj se «detiene» en la línea 320. Realmente, el reloj no puede detenerse; lo que tú haces es que la máquina recuerde una lectura particular en un instante determinado, por ejemplo cuando coinciden dos objetos en la pantalla.

El reloj está constituido por la variable TIME, que se encarga de llevar la cuenta del tiempo, y que se incrementa en una unidad, 50 veces por segundo.

TIME no para de incrementarse hasta que llega a 65535. Entonces vuelve a cero y empieza a contar otra vez.

Si nosotros escribimos TIME=0, como en la línea 75, la variable se pone a cero y empieza a contar. Al leer posteriormente la varible TIME, como hace la línea 320, el valor que leemos es el tiempo que ha transcurrido desde que pusimos a cero dicha variable. Así es como llevan la cuenta del tiempo la mayoría de los programas.

El reloj debe detenerse cuando el jugador ha rescatado a diez paracaidistas, por lo que en la línea 300 se comprueba si ya se han rescatado diez, en cuyo caso, el programa salta a la línea 320, que es la que «detiene» al reloj. La línea 340 imprime el tiempo que se ha tardado en rescatar a diez paracaidistas.

Si se ha rescatado con éxito a un paracaidista y, **además**, el número total de los que van ya rescatados es menor que diez, la línea 290 hace caer otro.

La línea 340 presenta el tiempo transcurrido para el rescate sólo si se han rescatado ya los diez. La lectura de tiempo es dividida por 50, por lo que el tiempo aparece en segundos. El reloj es actualizado 50 veces cada segundo.

MEJOR TIEMPO

De la misma forma que añadiste antes una opción de tanteo máximo para el juego, puede resultarte interesante una opción de mejor tiempo. En esta variante del juego tienes que poder registrar el tiempo más rápido en que los diez paracaidistas son rescatados, aunque este principio se puede aplicar a cualquier temporización que desees hacer.

He aquí las líneas que tienes que añadir:

```
20 LT=999999!
360 IF T<LT AND P=10
    THEN LT=T
380 LOCATE 0,21:PRINT
    "Mejor tiempo";LT
```

Igual que con el tanteo máximo se ponía inicialmente un «tanteo máximo» muy bajo, ahora se pone un «tiempo récord» ridículamente largo.

La línea 20 asigna a la variable de mejor tiempo (LT) un valor de 999999.

La línea 360 compara el último tiempo obtenido con el mejor tiempo. Si el último tiempo obtenido es más corto que el tiempo récord y, **además**, se han rescatado ya diez paracaidistas, entonces se modifica el tiempo récord haciéndolo igual al último tiempo obtenido.

Finalmente, la línea 380 sirve para expresar en segundos el tiempo récord. La variable de tiempo récord viene dividida por 50 para que resulte en segundos.

Una cosa que has de recordar es que si estás utilizando una opción de tiempo récord en un juego, tienes que utilizar la rutina de «¿Otra vez?», pues, de lo contrario, el valor del tiempo récord se perderá cada vez que ejecutes (con RUN) el programa.

EL TECLADO Y LA CUENTA DEL TIEMPO

Hasta ahora has visto cómo puedes controlar el reloj interno de la máquina desde dentro de un programa, examinando las posiciones de dos objetos sobre la pantalla. Otra forma de «detener» el reloj es servirte del teclado de tu ordenador.

Puedes hacerlo con la sentencia INKEY$. Resulta tan fácil como arrancar y parar un cronómetro para controlar el movimiento de los objetos por la pantalla.

Aquí tienes un juego de acción rápida que ilustra cómo puede usarse el teclado para detener el reloj:

```
20 CLS
30 A=RND(-TIME):N=INT
   (RND(1)*900)+1
40 FOR F=0 TO N
50 NEXT F
55 FOR F=1 TO 50:C$=INKEY$
   :NEXT F
60 CLS:LOCATE 0,10:PRINT
   "Dispara!!"
70 TIME=0
80 A$=INKEY$:IF A$="" THEN
   80
90 T=TIME
100 LOCATE 0,10:PRINT
    "BANG!!!!!"
110 FOR F=1 TO 300
120 NEXT F
130 M=INT(RND(1)*35)+1
140 IF T<M THEN LOCATE 0,15
    :PRINT"Has sobrevivido"
150 IF T>M THEN LOCATE 0,15
    :PRINT"Has muerto      "
160 IF T=M THEN LOCATE 0,15
    :PRINT"Habeis muerto   "
```

El programa presenta el mensaje «DISPARA!!» y el jugador ha de pulsar cualquier tecla tan rápido como pueda. Se mide el tiempo de reacción desde el momento en que apareció el mensaje.

Las líneas 30 a 50 introducen una pausa aleatoria. La línea 60 sirve para enviar el mensaje «DISPARA!!» e inmediatamente se arranca el contador de tiempo en la línea 70. La línea 80 hace que la máquina espere, continuando cuando se ha pulsado una tecla cualquiera. Ya vimos esta línea al ocuparnos del «Control del Teclado».

En cuanto se ha pulsado una tecla cualquiera, la línea 90 para el contador, llamado T a la lectura que tiene en ese momento. La línea 100 escribe «BANG!!». Hay una pausa introducida por las líneas 110 y 120 antes de que la máquina elija un instante de tiro. La línea 130 es la que se encarga de hacer esto.

La máquina tiene ahora dos variables, su tiempo, T, y el tiempo de la máquina, M. Las líneas 140 a 160 comparan estos valores y presentan el resultado del duelo.

P y R

¿Existe algún límite para la duración máxima que se puede tener?

Sí existe un límite, aunque normalmente es tan alto que en la práctica no tiene importancia. El reloj interno de casi todos los ordenadores domésticos avanza a la misma velocidad, y el factor limitador es la cantidad de pulsos de tiempo que el ordenador puede recordar.

En tu **MSX** se puede contar hasta dos bytes (65535) lo que equivale aproximadamente a unos 22 minutos.

Foto de la pantalla de un momento del juego.



JUEGOS DE LABERINTO

- LOS PRINCIPIOS DE LA ANIMACION
- MOVIMIENTO DE GRAFICOS
- COMO UTILIZAR LOS GRAFICOS INCORPORADOS

Los juegos de laberintos sofisticados requieren programas largos. Pero tú puedes diseñar algunos sencillos extrayendo de ellos importantes principios y utilizando poco más que un bucle y sentencias DATA.

Los juegos de laberintos ejercen una fascinación permanente sobre los propietarios de un ordenador, por lo que las casas de *software* continúan sacando nuevas variantes del comecocos.

Este artículo te enseñará la manera de saltar al carro de los fabricantes de laberintos, permitiendo que te construyas el tuyo propio.

En su primera fase el laberinto no incluye «enemigos» ni obstáculos, ya que esto requeriría un programa muy largo. Pero te enseñará cómo se programa el que tu carácter principal no pueda atravesar las paredes, lo cual es la base de todos los juegos de esta clase. También se incluye la puntuación y el crono, así como una rutina de «mejor tanteo», a fin de darle un cierto interés competitivo.

La manera más fácil de entender cómo funciona el juego del laberinto es ir introduciéndolo por etapas. Así pues, empieza construyendo el propio laberinto:

```
100 CLS:FOR N=3 TO 17
110 READ A$
120 FOR M=7 TO 21
130 LOCATE M,N:PRINT"."
140 IF MID$(A$,M-6,1)="p"
    THEN LOCATE M,N:PRINT
    CHR$(219)
150 NEXT M
160 NEXT N
9000 DATA "ppppppppppppppp"
9010 DATA "p.............p"
9020 DATA "p.pp.pp.pp.pp.p"
9030 DATA "p.p.........p.p"
9040 DATA "p...p.p.p.p...p"
9050 DATA "p.ppp.p.p.ppp.p"
9060 DATA "p.....p.p.....p"
9070 DATA "pppp.pp.pp.pppp"
9080 DATA "p.....p.p.....p"
9090 DATA "p.ppp.p.p.ppp.p"
9100 DATA "p...p.p.p.p...p"
9110 DATA "p.p.........p.p"
9120 DATA "p.pp.pp.pp.pp.p"
9130 DATA "p.............p"
9140 DATA "ppppppppppppppp"
```

Las líneas 100, 120, 150 y 160, que definen un par de bucles FOR ... NEXT, establecen los contornos del laberinto. Con la línea 130 se imprime un punto en cada cuadrado.

Las líneas 110 y 140 se encargan de leer los datos de las líneas 9000 a 9140 y de sustituir el punto por un bloque macizo, que es el carácter que corresponde al código **ASCII** 219, cada vez que en la sentencia DATA correspondiente aparece la letra p. De esta forma se obtiene un laberinto de paredes macizas en el que los pasillos interiores están cubiertos de puntos. Como podrás ver cuando teclees RUN se trata del clásico laberinto de los programas del tipo **comecocos**.

CONSTRUYENDO EL «COMILON»

Sin embargo, un laberinto resulta bastante inútil si no hay algo que se mueva a través de él. Ejecuta pues (con RUN) el programa, y añádele lo siguiente:

```
40 WIDTH 40
50 X=14
60 Y=10
```





```
1000 LOCATE X,Y:PRINT "*"
1010 XX=X
1020 YY=Y
1030 B=STICK(0):IF B=0
     THEN 1030
1040 IF B=1 AND VPEEK(40*
     (Y-1)+X)<>219 THEN Y=Y-1

1050 IF B=5 AND VPEEK(40*
     (Y+1)+X)<>219 THEN Y=Y+1

1060 IF B=3 AND VPEEK
     (40*Y+X+1)<>219 THEN
     X=X+1
1070 IF B=7 AND VPEEK
     (40*Y+X-1)<>219 THEN
     X=X-1
1080 LOCATE XX,YY:PRINT" "
1090 GOTO 1000
```

Como ya has experimentado con el control de movimientos mediante las teclas de cursor, la mayor parte de estas nuevas líneas te resultarán familiares. Lo que hacen es colocar un **comecocos** (en este caso se trata de un sencillo asterisco) en el interior del laberinto, haciendo que se mueva en la dirección que tú le indiques mediante las teclas de cursor.

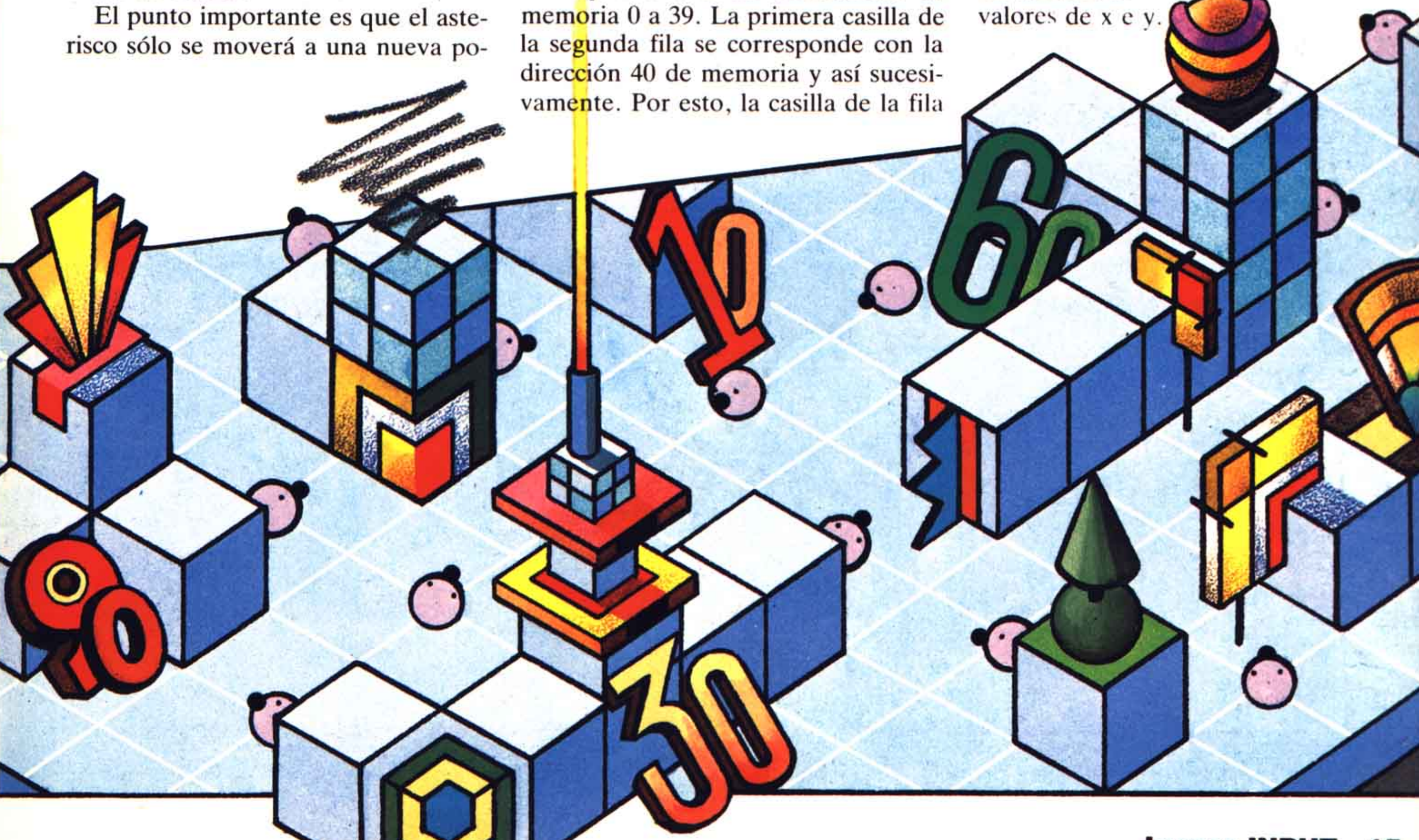
El punto importante es que el asterisco sólo se moverá a una nueva posición si el muro del laberinto se lo permite. Para que lo entiendas, imagina que el asterisco está situado en el laberinto en las coordenadas x,y. Si quieres desplazarlo una casilla hacia arriba tendrás que restarle uno a la coordenada y, situando el asterisco en las nuevas coordenadas x, y−1. Pero esto sólo podrás hacerlo si en estas nuevas coordenadas no hay muro. Para saber si esto ocurre, el programa utiliza la instrucción VPEEK, que permite leer directamente los contenidos de la memoria de pantalla (la famosa **VRAM**).

En próximos números de **INPUT** comentaremos más a fondo esta instrucción y la forma de manejar con ella la **VRAM**, pero por el momento nos basta con saber que VPEEK(40*y+x) proporciona el valor **ASCII** del caracter que se encuentra en las coordenadas x,y de la pantalla. Esto es así porque la memoria de pantalla se organiza en direcciones consecutivas de memoria que representan las casillas de la pantalla. Como estamos trabajando en 40 columnas, la primera fila de la pantalla se corresponde con las direcciones de memoria 0 a 39. La primera casilla de la segunda fila se corresponde con la dirección 40 de memoria y así sucesivamente. Por esto, la casilla de la fila y, columna x, tiene como dirección de memoria 40*y+x.

Entre las líneas 1030 y 1070 el programa se encarga de ver si se ha pulsado alguna tecla de cursor. Si ha sido así, mueve el asterisco a la nueva posición, pero sólo tras comprobar que el caracter que ocupa dicha posición no tiene el código **ASCII** 219, es decir, no es el muro.

Por su parte, las líneas 1010, 1020 y 1080 definen la posición que el asterisco acaba de abandonar y se encargan de borrar los puntos comidos, poniendo un espacio en blanco mediante PRINT. Estas líneas son muy importantes en cualquier programa de este tipo y se encargan de algo que, aunque parece trivial, hay que tener siempre muy en cuenta.

Se trata de que al cambiar el asterisco de la posición x,y a una nueva posición, que también se llama x,y, hay que guardar en algún sitio los anteriores valores x,y, para que el programa sepa donde estaba antes el asterisco y pueda borrarlo. Para ello este programa utiliza las variables xx,yy en las que se guardan temporalmente los anteriores valores de x e y.



PUNTUACION Y CRONOMETRAJE

Ya tienes todos los elementos para construir un juego sencillo. Para que de verdad se pueda jugar con él, y en ausencia de «enemigos» —que harían el programa desmedidamente largo— lo mejor es incluir una rutina de cronometraje y tanteo. Agrega pues las siguientes líneas:

```
10 BT=100000!
20 PUN=0
30 TIME=0
1085 IF VPEEK(40*Y+X)=46
     THEN PUN=PUN+1
1087 LOCATE 0,20:PRINT
     "Puntos ";PUN
1090 IF PUN=110 THEN
     T=TIME:GOTO 2000
1095 GOTO 1000
2000 LOCATE X,Y:PRINT" "
2010 LOCATE 0,20:PRINT
     "Tiempo empleado:";
     T/50;"Segundos"
2020 IF T<BT THEN BT=T
2030 LOCATE0,21:PRINT
     "Mejor tiempo";BT/50
```

El sistema de puntuación es bastante sencillo. La línea 20 pone a cero la puntuación al comenzar la partida. La línea 1085 incrementa en uno la puntuación cada vez que el asterisco se come un punto. Para ello se comprueba, mediante VPEEK, si en la casilla en que se coloca el asterisco había un punto (el código **ASCII** del punto es 46) o no lo había.

La línea 1087 se encarga de imprimir la puntuación en la parte inferior de la pantalla.

Cuando el asterisco se ha comido todos los puntos la puntuación vale 110. Entonces, la línea 1090 lee el tiempo que ha transcurrido y salta a la sección de cronometraje, en la línea 2000.

Esta sección es también muy fácil de entender. Empieza en la línea 10, fijando el «mejor tiempo» inicial en 100000, mucho más de lo que hará cualquier jugador.

A continuación, en la línea 30, se pone el reloj a cero con lo que da comienzo la cuenta del tiempo. Este transcurre segundo a segundo, hasta que el jugador se «come» todos los puntos del laberinto y consigue con ello que la puntuación llegue a 110. En ese momento y en la línea 1090 se lee el tiempo transcurrido. Entnces se pasa a la línea 2000, que se encarga de borrar al asterisco, y luego a la 2010 que imprime el tiempo empleado, en segundos. Como el reloj cuenta cincuentaavos de segundo, se divide por cincuenta y se obtienen segundos.

Por último, las líneas 2020 y 2030 comparan el tiempo obtenido con el «mejor tiempo». Si el tiempo obtenido es mejor, queda registrado como nuevo tiempo. En la línea 2030 se imprime este mejor tiempo, en segundos.

OTRA VEZ

Para darle al jugador otra oportunidad, tienes que introducir las siguientes líneas:

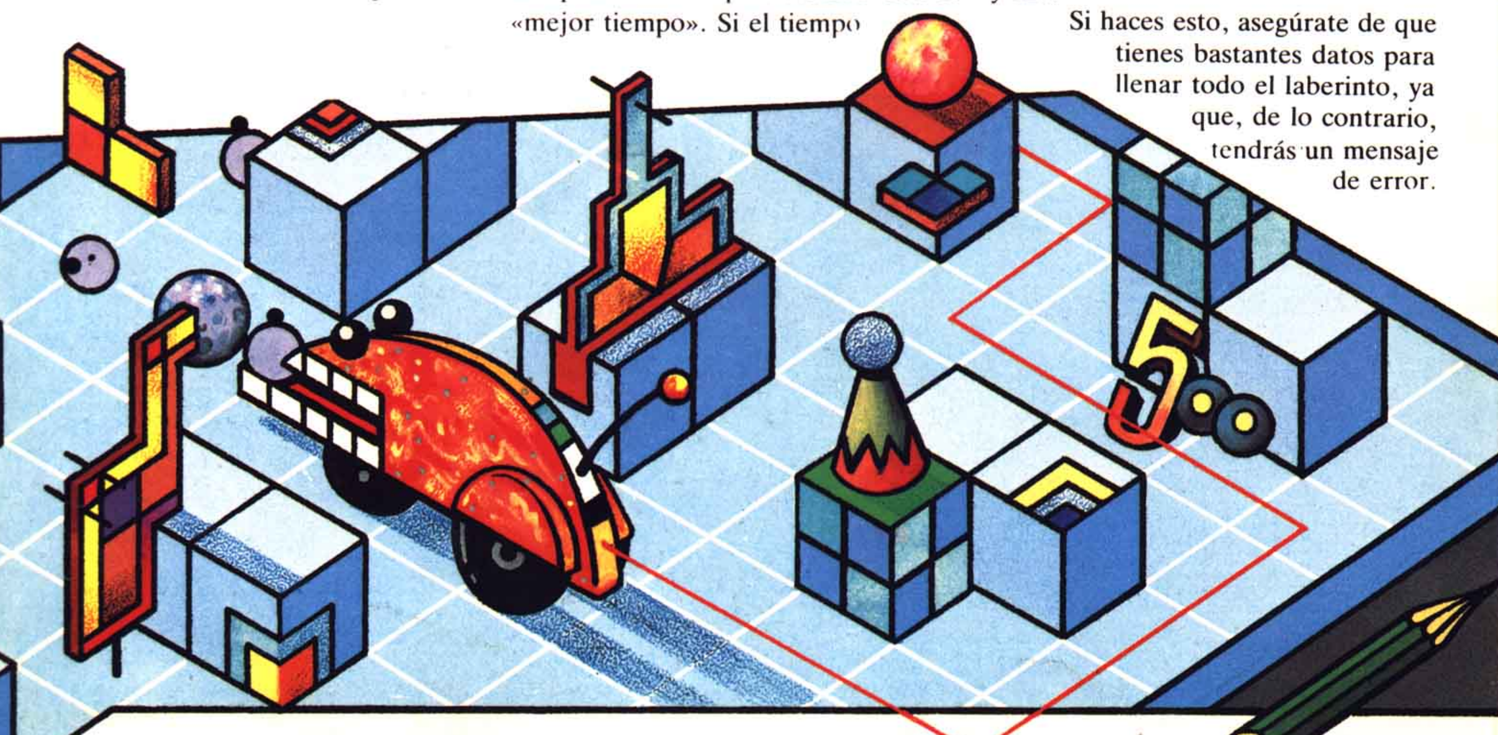
```
2040 FOR J=1 TO 1000:NEXT
2050 LOCATE0,22:PRINT
     "Otra vez ? (S/N)"
2060 C$=INKEY$:IF C$=""
     THEN 2060
2070 IF C$="s" THEN RESTORE
     :GOTO 20
2080 IF C$="n" THEN CLS:END
2090 GOTO 2060
```

Con RESTORE se vuelve al principio de la lista de los DATA.

OTROS LABERINTOS

Si quieres probar otros laberintos con las mismas dimensiones que éste, puedes hacerlo simplemente cambiando la disposición de las letras p en las sentencias DATA, desde la línea 9000 en adelante. Para tener un laberinto más grande o uno más pequeño, también tienes que volver a definir los límites o contornos, cambiando los números que aparecen en las líneas 100 y 120.

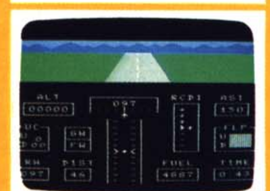
Si haces esto, asegúrate de que tienes bastantes datos para llenar todo el laberinto, ya que, de lo contrario, tendrás un mensaje de error.



microbyte PRESENTA... MSX

SENSACIONALES PROGRAMAS EN CARTUCHO Y CASSETTE

FLIGHT PATH 737.



Colócate a los mandos de un jet comercial. Disponemos de control total sobre los mandos del avión, y puedes escoger entre 6 niveles de dificultad.

P.V.P.: CART. 3.490 pts.
CASS. 1.900 pts. 32K.

FRUITY FRANK



Tu jardín ha sido inválido por monstruos de fruta madura. La única forma de combatirlos es lanzarles fruta fresca del jardín.

P.V.P.: CASS. 1.900 pts. 32K.

SPARTAN X



Son muchos los peligros que te acechan. Ten los reflejos bien despiertos, pon tus fuerzas en estado de alerta, y a luchar.

P.V.P.: CASS. 1.900 pts. 32K.

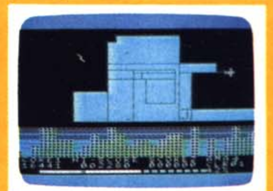
CHUCKIE EGG



Debes recoger los huevos antes de que nazcan los pollitos y se coman el maíz. Pero ojo con el Pato Loco.

P.V.P.: CASS. 1.900 pts. 32K.

NIGHT FLIGHT



Con tu pequeño avión, debes ir dando luz a la noche, hasta que el cielo esté de nuevo azul. Date prisa en realizar tu misión, de lo contrario...

P.V.P.: CART. 2.900 pts.
CASS. 1.900 pts. 16K

STAR AVENGER



Imagina el juego de batalla más rápido que jamás hayas visto. Piensa además, en los más excitantes gráficos y sus 5 niveles de dificultad. Todo ello es Star Avenger.

P.V.P.: CASS. 1.900 pts. 32K.

GYRO ADVENTURE



Ponte a los mandos de tu helicóptero y combate a los enemigos que se enfrentan a tí. Podrás mover el helicóptero en todas direcciones, mantenerlo en el aire y disparar. **P.V.P.: CART. 2.900 pts. CASS. 1.900 pts. 16K.**

SUPER CROSS FORCE



Sólo queda una esperanza para la supervivencia ante el ataque de los malvados Morpul. Tú podrás atacarles, con tus naves dispuestas en paralelo o en diagonal.

P.V.P.: CART. 2.900 pts. 16K.

JUMP LAND



Tu mayor obsesión han sido siempre los pasteles, y por ello, te has visto envuelto en situaciones complicadas que has salvado gracias a tus reflejos.

P.V.P. CASS. 1.900 pts. 16K.

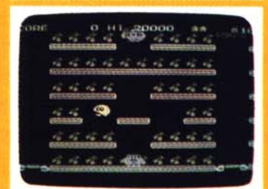
ROGER RUBBISH



Los perversos contaminadores de planetas están llenando nuestra galaxia de residuos nucleares. Roger Rubbish es el más famoso recogedor de basuras espaciales.

P.V.P.: CART. 2.900 pts. 16K.

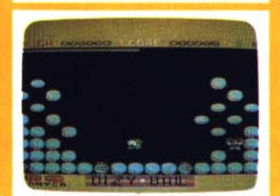
FRUIT PANIC



Un día, Walky, para divertirse se fué al país de los gatos. ¿Cuánta fruta podrá comerse Walky?

P.V.P.: CASS. 2.000 pts. 16K

DIZZY BALLOON



En este mundo hay seres voladores y atacan cuerpo a cuerpo. Si los haces explotar, se irá abriendo el cielo y tendrás la oportunidad de escapar.

P.V.P.: CASS. 2.000 pts. 32K.

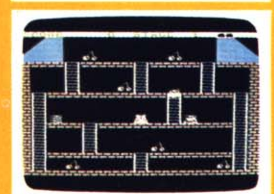
CASTLE COMBAT



El castillo galáctico, ha caido bajo la dominación de los Tyrones. Tu nave STAR DUSTER, está preparada para el combate. ¿Te atreves?

P.V.P.: CART. 2.900 pts. 16K.

NICK NEAKER



Cuando estás dormido, muchas cosas suceden a tu alrededor. Algunos objetos de tu casa toman vida, como en el caso de la zapatilla NICK.

P.V.P.: CART. 2.900 pts.
CASS. 1.900 pts. 16K.

CHAMP



Champ es un completo Ensamblador/Monitor para tu MSX. Champ te permite escribir y trazar programas en código máquina con el mínimo esfuerzo.

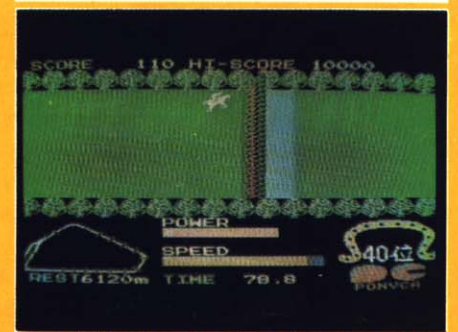
P.V.P.: CART. 3.890 pts.
CASS. 2.400 pts. 32K.

KARATE



Has conseguido entrar en la cueva de los piratas y ahora comienzan tus problemas. Los murciélagos gigantes, moradores de estas cuevas pueden chuparte la sangre. Cuando te encuentres con los piratas, deberás enfrentarte a ellos con tu depurado estilo de Kárate. **P.V.P.: CART. 3.490 pts. CASS. 1.900 pts. 32K.**

GRAND NATIONAL



Si te gustan las carreras de caballos, no te quedes como un espectador, participa. Ahora puedes correr con tu caballo, en la más prestigiosa carrera del mundo, el GRAND NATIONAL.

P.V.P.: CASS. 2.000 pts. 32K.

ENVIENOS A **MICROBYTE**
P.º Castellana, 179, 1.º - 28046 Madrid

Nombre
Apellidos
Dirección
Población
D.P. Teléfono

ENVIOS GRATIS

JUEGO	Cart.	Cass.	Precio	TOTAL

PRECIO TOTAL PESETAS

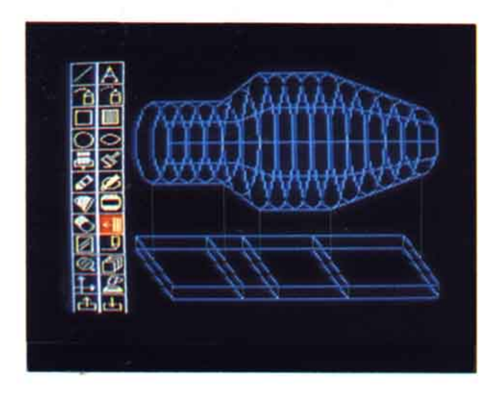
Incluyo talón nominativo ☐
Contra-Reembolso ☐

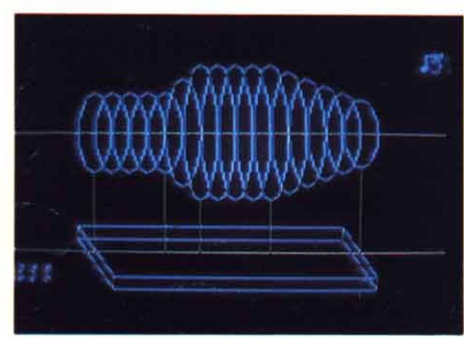
Pedidos por teléfono **91 - 442 54 33 / 44**

EL RATON Y EL QUESO

Hasta ahora, para dibujar, no había más remedio que coger la caja de pinturas, unas cuantas láminas de dibujo y, armándose de valor y teniendo por delante una buena cantidad de tiempo, acometer la labor de transformar la lámina en blanco en toda una obra maestra.

¿Ha cambiado algo en plena era de la revolución informática?. Quizá algo sí, aunque no mucho todavía. El papel y la caja de pinturas siguen siendo los elementos más utilizados pero... Poco a poco comienzan a verse, en los lugares más insospechados, imágenes y dibujos realizados con ayuda de un ordenador. ¿Quién no se ha admirado con las extrañas y un tanto irreales imágenes que aparecen en algún *spot* de televisión? ¿Y qué decir de algunas películas, como por ejemplo la que lleva el título de **TRON**, cuyas imágenes han sido generadas íntegramente mediante un potente ordenador?





A medida que el desarrollo de la microelectrónica ha ido abaratando los costes de las memorias y de los procesadores especializados y a medida que estos han conseguido mayor velocidad y mayor capacidad de proceso, han llegado a ser una realidad los programas de diseño gráfico, capaces de servir de ayuda, e incluso de fuente de inspiración, en la creación de los más increibles dibujos.

Y, cómo no, algo de esto ha llegado hasta el ordenador doméstico a través de programas adaptados a las escasas capacidades de estas máquinas (si las comparamos con las de sistemas profesionales específicos), pero que conservan buena parte de la filosofía de los grandes programas de diseño asistido por ordenador.

El paquete gráfico **NEOS MS-10**, para **MSX** es un buen ejemplo de ello.

EL RATON Y EL QUESO

NEOS MS-10 está formado por dos elementos; por una parte el *hardware*, constituido por un pequeño ratón, periférico de acción similar a la de *joysticks* y *paddles* pero con características propias que lo convierten en un elemento de comunicación usuario-ordenador ideal para aplicaciones gráficas. Y por otra parte el *software*, un programa de diseño gráfico que ha sido bautizado con el apropiado nombre de **Cheese** (en castellano significa queso) y que se suministra en cartucho ROM o en *cassette*.

La combinación de estos dos elementos, como vamos a ver, da como resultado un sistema potente, flexible y agradable para el usuario, que permitirá realizar, con un poco de práctica, cualquier dibujo que a uno se le pase por la imaginación.

El programa ofrece una serie de posibilidades para trabajar en la pantalla gráfica de alta resolución. Dichas posibilidades se representan en un menú gráfico, cada opción mediante un dibujito (**icono**). Para seleccionar una u otra no hay más que recurrir al ratón. Al desplazarlo sobre una superficie plana veremos como, simultáneamente, se desplaza un rectángulo de color rojo a través de los dibujos del menú. Una vez que se ha elegido opción, es decir cuando el icono que la representa aparece sobre fondo rojo en lugar de sobre fondo negro, no hay más que pulsar el botón izquierdo del ratón para que la opción entre en funcionamiento.

Cada una de las opciones ofrece una dinámica posterior diferente, pero en cualquier caso el manejo siempre será mediante movimientos del ratón y pulsaciones sobre su botón izquierdo. El otro botón, el derecho, está reservado para la vuelta al menú desde no importa que situación o posición.

Sólo hay un menú que engloba todas las funciones; las que son puramente de diseño y las que no lo son, como por ejemplo las de transferencia de imágenes entre la memoria y el *cassette*. Esto simplifica bastante las cosas.

Para dar nuestras impresiones sobre el conjunto ratón-programa, no nos quedaba más solución que ponernos en la piel del dibujante y acometer la tarea de dibujar algo. Esto es lo que hicimos...

COMO HACER UN DIBUJO

Después de unas cuantas horas de ensayos y pruebas, nos decidimos a dibujar el paisaje de una bahía. Al trabajar sobre el paisaje fuimos descubriendo unas cuantas cosas sobre el programa.

Nuestro dibujo empezó como un boceto en el que representamos las lí-



neas de la costa. Para ello seleccionamos el pincel más fino, el color gris y la opción para dibujar líneas. Con esta opción se puede dibujar cualquier tipo de línea curva. Al mover el ratón, el lápiz de la pantalla dibuja una curva que sigue exactamente el movimiento del ratón. Si se mantiene pulsado el botón, el lápiz pinta, si deja de pulsarse, el lápiz deja de pintar. Después de dibujar los contornos de nuestro paisaje empezamos a dar color. Para ello utilizamos varias de las opciones del menú. La primera de ellas fué utilizar el pincel mas grueso. Este se comporta casi como una brocha, (tiene una anchura de trazo de cuatro *pixels*) permitiéndonos dar verdaderos brochazos de color. La opción resultaba muy adecuada para dar color a zonas de pequeño y mediano tamaño, pero se quedaba corta para las zonas de mayor extensión. En ellas resultó mucho más adecuada la opción de relleno de superficies, que permite llenar de color áreas limitadas por una línea cerrada que, eso sí, tiene que ser del mismo color que el que se utilice en el interior. Si no tenemos en cuenta este pequeño detalle, como nos ocurrió más de una vez, al terminar de llenar el interior, la pintura se saldrá y manchará diversas zonas del dibujo. Para el cielo de nuestro paisaje, nos decidimos a utilizar tramas, que son combinaciones de dos colores primarios. En la zona coloreada con una trama, las líneas horizontales de *pixels* van alternando entre los dos colores. El efecto de tener una línea de cada color es prácticamente el de tener un color distinto. Hay un total de 120 tramas diferentes que proporcionan bastante flexibilidad en la elección de los colores.

Al llenar de color las distintas zonas es inevitable que aparezcan una serie de defectos, relacionados tanto con la falta de pulso del dibujante como con las restricciones al número de colores, por bloque de 8×8 *pixels*. Estas imperfecciones tuvimos que eliminarlas en una fase posterior. Para ello teníamos a nuestra disposición la opción lupa, que aumenta el tamaño de una zona del dibujo. Esta zona aumentada aparece en la parte superior de la pantalla con lo que se puede trabajar sobre ella modificando individualmente cada *pixel*, cambiándolo de color.

Esta opción y utilizar el pincel más fino, nos permitieron retocar nuestro dibujo hasta darle el acabado que puede verse en las fotografías de pantalla.

Después de los ensayos previos que habíamos hecho y con la práctica adquirida, realizar el dibujo nos ocupó escasamente una hora. ¡No está mal para unos dibujantes sin experiencia!

Aunque muy contentos con nuestro paisaje, nos quedaban ganas de seguir experimentando. Así, nos decidimos a probar suerte y tratamos de hacer un dibujo mucho más geométrico. De este modo íbamos a tener ocasión de utilizar las opciones de diseño de líneas rectas, circunferencias, elipses y rectángulos.

Teníamos la intención de dibujar una figura de revolución y probamos con la opción de dibujo de circunferencias. Cada vez que intentábamos dibujar una circunferencia, ante nuestros ojos aparecía una elipse. La explicación está en que los *pixels* que aparecen en la pantalla no son cuadrados; son rectangulares. De esta forma, aunque el radio de la circunferencia, medido según la horizontal, tenga el mismo número de *pixels* que el medido según la vertical, el primero resulta de mayor longitud al ser los pixels más anchos horizontal que verticalmente. Este defecto, común en muchos sistemas de diseño gráfico, podría haberse corregido en origen, de modo que al dibujar nuestra circunferencia obtuviéramos realmente una circunferencia. Ante esta eventualidad, nos decidimos a utilizar elipses. Con gran rapidez dibujamos el esqueleto de nuestra figura de revolución. Pudimos hacerlo con absoluta precisión gracias a una característica muy interesante de **Cheese** y es que, en la esquina inferior izquierda de la pantalla, aparecen en todo momento las coordenadas x e y del *pixel* que estamos dibujando. De esta forma es fácil dibujar una serie de elipses sobre un eje común, rectas paralelas que verdaderamente lo sean, medir distancias sobre el dibujo, etc.

Pasamos después a la opción para dibujar rectas con la que completamos nuestro dibujo. Habíamos tardado bastante menos que al dibujar el paisaje costero.

NUESTRAS IMPRESIONES

Después de dibujar durante bastantes horas, llegamos a apreciar las cosas buenas y también los defectos del ratón y del queso. Descubrimos que el primero resulta un elemento ideal para dibujar, incluso mejor que una tableta gráfica (aunque esto es cuestión de gustos). El ratón es sobre todo veloz, sensible y preciso. Se mueve por la pantalla a unas velocidades im-



Los Joysticks más

QUICKSHOT IV
(3 en 1) Con mando
para deporte

QUICKSHOT IV (3 en 1)
Con mando de carreras

QUICKSHOT I MSX

QUICKSHOT I

QUICKSHOT VII - Portátil

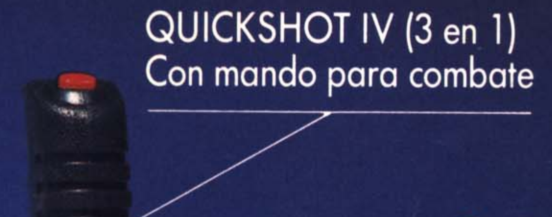
QUICKSHOT IX
Preciso y sensible

Los QUICKSHOT comercializados por SVI-España, S. A. son los únicos que tienen la GARANTIA OFICIAL SVI.



vendidos del mundo.





QUICKSHOT IV (3 en 1)
Con mando para combate

QUICKSHOT II MSX
Con autodisparo



QUICKSHOT II
Con autodisparo



QUICKSHOT VII MSX
Portátil

Importador exclusivo SVI-España.



presionantes respondiendo al menor movimiento de la mano, y es capaz de llegar sin problemas a cualquier *pixel* y de pasar de un *pixel* al siguiente sin el menor titubeo. El ratón necesita de una superficie plana sobre la que deslizarse. Lo más sencillo es utilizar la propia mesa en la que se encuentra el ordenador, basta con dejar un poco de espacio libre de objetos, en realidad muy poco, digamos unos 20×20 centímetros. Al principio resulta un tanto difícil hacerse con el manejo del ratón, pero a medida que se experimenta se van apreciando todas sus posibilidades.

Cheese es otra historia totalmente distinta. Es un excelente programa con una buena cantidad de opciones para dibujar, a las que se accede de forma sencilla desde un único menú. El color es uno de los puntos fuertes del programa. Se pueden utilizar los 16 colores distintos del **MSX** y además, un total de 120 tramas bicolores, que se comportan casi como colores diferentes. La única queja que tenemos con respecto al color es que no se haya incluido la posibilidad de superponer una rejilla que delimite los bloques de 8×8 *pixels*, en los que existen restricciones de color. A veces resulta exasperante tener un dibujo bastante avanzado y de repente caer en la cuenta de que no se puede utilizar cierta combinación de colores porque entran en conflicto dentro de un bloque de 8×8.

Las posibilidades para el trazado de figuras geométricas son quizá las más flojas de todo el conjunto. Especialmente pobre es el trazado de líneas rectas. No se ha contemplado la posibilidad de definir un centro a partir del que radien las rectas, ni la facilidad para encadenar rectas de modo que el final de una, sea al mismo tiempo principio de la siguiente. Estas pequeñas pegas no son de importancia, se pueden obviar con un poco más de trabajo por parte del dibujante, pero no dejan de deslucir un tanto la brillantez del paquete gráfico.

La opción lupa, aunque correcta resulta un tanto incómoda de utilizar. Al escoger dicha opción, en la parte superior de la pantalla aparece un recuadro por el que va desfilando, ampliada, la parte de imagen sobre la que coloquemos el cursor. Resultaría mucho más agradable trabajar con una lupa en la que la imagen ampliada permaneciera fija.

También se echa de menos una opción que permitiera borrar o deshacer la última acción llevada a cabo. Por ejemplo, cuando se traza una línea por equivocación no queda más remedio que borrarla punto a punto, lo que puede resultar tedioso si la línea en cuestión está en una zona comprometida del dibujo y hay que tener cuidado para no borrar los alrededores de la línea.

Muy bien realizada está, sin embargo, la opción para definir ejes de simetría, también conocida como opción de espejo. Al contrario que en otros programas de diseño gráfico, en los que sólo se puede optar por un eje de simetría, en **Cheese** es posible utilizar hasta cuatro simultáneamente (uno vertical, otro horizontal y dos diagonales).

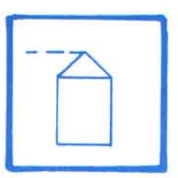
Otra de las opciones que más nos gustó es la que sirve para cambiar un color por otro. La opción funciona de la siguiente manera: imaginemos que al dibujar el paisaje costero se nos

CHEESE: LAS OPCIONES DEL MENU

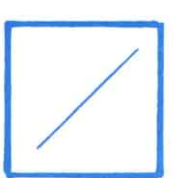
COMANDOS DE DISEÑO



Permite dibujar líneas que reproducen el movimiento del ratón. Al mantener pulsado el botón izquierdo la línea aparece en pantalla, al soltar, el cursor se mueve pero sin dibujar.



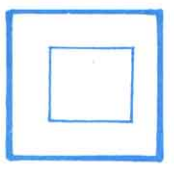
Igual que la opción anterior, pero dibujando líneas de trazos, es decir un *pixel* sí otro no.



Líneas rectas. Con el ratón si fijan los dos extremos del segmento.



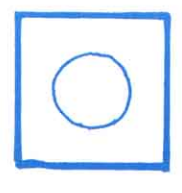
Caracteres. Se introducen desde el teclado y aparecen en la posición que ocupa el cursor.



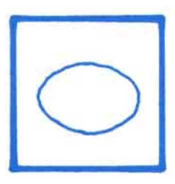
Rectángulos. Con el ratón se fijan los extremos de una de las diagonales del rectángulo. Al pulsar el botón el rectángulo queda dibujado.



Rectángulos coloreados. Igual que la anterior, pero en este caso el interior del rectángulo aparece coloreado en el color elegido.

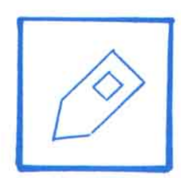


Círculos. Con el ratón se escoge el centro del círculo y el radio del mismo. Al ser los *pixels* rectangulares en lugar de cuadrados, los círculos aparecen deformados; el diámetro horizontal es mayor que el vertical.

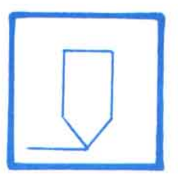


Elipses. Con el ratón se escogen sucesivamente el centro, el diámetro horizontal y el diámetro vertical de la elipse.

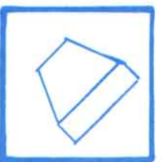
COMANDOS DE SELECCION



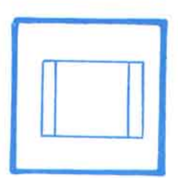
Color del lápiz. Al pulsar aparece una paleta con los 16 colores entre los que se puede escoger.



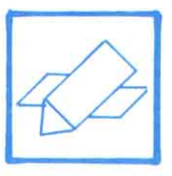
Grosor del lápiz. Hay tres grosores posibles. El más fino tiene una anchura de un pixel, el intermedio de dos y el grueso de cuatro.



Color de fondo. A seleccionar entre 16 colores.

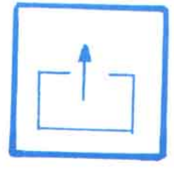


Color del borde. Corresponde a dos franjas, una por encima y otra por debajo del fondo.

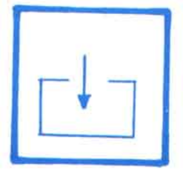


Color del cursor. En la pantalla aparece un cursor con forma de lápiz. Con esta opción es posible cambiarlo de color para destacarlo del fondo. Por ejemplo, sobre un fondo de color negro, utilizaremos un color claro para el cursor.

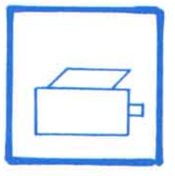
COMANDOS DE ENTRADA Y SALIDA



Load. Sirve para cargar en memoria un dibujo previamente almacenado en cassette.

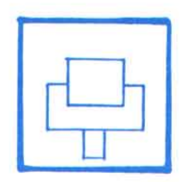


Save. Se utiliza para almacenar un dibujo en cassette.

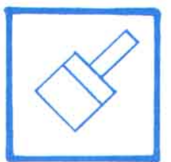


Print. Permite realizar un volcado de pantalla a través de una impresora matricial, ya sea en blanco y negro o en color.

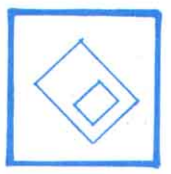
COMANDOS DE MANEJO DEL COLOR



Llenado de superficies. Colorea una superficie cerrada con el mismo color que el de la línea que la encierra.

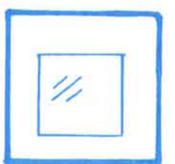


Tramado. Permite rellenar cualquier superficie cerrada con una trama bicolor. Hay hasta 120 tramas distintas.

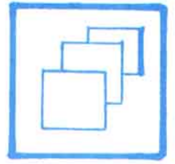


Cambio de color. Permite cambiar, en una zona rectangular del dibujo, un color determinado por otro cualquiera.

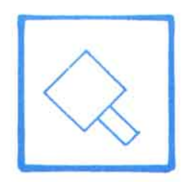
COMANDOS ESPECIALES



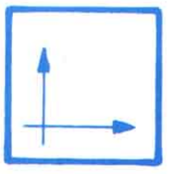
Espejo. Ofrece la posibilidad de trabajar con hasta cuatro ejes de simetría sobre la pantalla. Los ejes son: uno vertical, uno horizontal y dos diagonales.



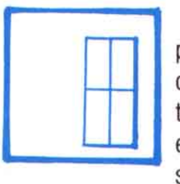
Copia. Esta opción da la posibilidad de copiar cualquier porción rectangular de un dibujo en otra zona, distinta de la original, dentro del mismo dibujo.



Lupa. Aumenta la porción del dibujo que se encuentra bajo el cursor. Se utiliza para los pequeños detalles y permite trabajar *pixel* a *pixel*.



Scrolling. Esta opción permite desplazar el dibujo hacia arriba, abajo, a la izquierda o a la derecha de la pantalla.



Situación del menú. Sirve para colocar el menú en cualquier posición de la pantalla. Con ello se evitará que el menú estorbe el trabajo sobre el dibujo.

ocurre que, en una zona de la montaña, estaría mejor cambiar el verde claro por otro color, por ejemplo un verde más oscuro. Para ello, volvemos al menú, escogemos la opción de cambio de colores, y al aparecernos la paleta con todos los colores disponibles, elegimos primero el verde claro y luego el verde oscuro (primero el color original y luego el color final). Hecho esto nos encontraremos de nuevo en el dibujo, sobre el que indicaremos la zona en la que deseamos efectuar el cambio. Para ello encerraremos la zona dentro de un rectangulo utilizando el ratón. Ahora no nos queda más que pulsar el botón y ¡zas!, veremos como todos los *pixels* de color verde claro cambian a verde oscuro. Esta opción tan sencilla, encierra gran parte de la filosofía de los programas de diseño por ordenador. Con sólo pulsar un botón, es posible cambiar los colores de un dibujo, instantáneamente, sin tener que volver a dibujar. Esto significa que es muy sencillo y rápido experimentar con un dibujo, es decir, responder casi al instante a preguntas del tipo: ¿Y cómo quedaría el dibujo si...? En un futuro próximo, es más que probable que todo dibujante profesional cuente con un sistema de este tipo. De esta forma, al plantearse un dibujo, el dibujante podrá experimentar rápidamente varias posibilidades, escoger la más agradable y luego pasar a la realización del dibujo en sí, pero ya con las ideas bastante claras de cual puede ser el resultado final.

Además, si en cualquier momento, durante la realización del dibujo, se le ocurren nuevas ideas, no tendrá más que coger su ratón, moverlo unas cuantas veces y ver si las ideas eran buenas, sólo aceptables o francamente malas. El ahorro de tiempo puede ser considerable.

Por otra parte, estamos hablando de un sistema sencillo, como **Cheese**, diseñado para los **MSX** de la primera generación que, aunque cuentan con unas capacidades gráficas que hubieran resultado impensables hace unos pocos años, estas capacidades son muy inferiores a las de por ejemplo los **MSX** de la segunda generación, que cuentan con más resolución, mayor número de colores, más velocidad en el tratamiento de las imágenes y otras muchas sorprendentes capacidades.

Esperamos hablaros muy pronto de las características gráficas de estas nuevas máquinas. Haremos una comparación entre las dos generaciónes y comentaremos los programas de diseño de **MSX2**.

Cheese y el ratón nos han parecido un estupendo producto. La mejor herramienta para dibujar y para sacar el mejor partido de las posibilidades gráficas del ordenador. Para no extendernos más y teniendo en cuenta que una imagen vale más que mil palabras os recomendamos que miréis detenidamente las fotografías que acompañan a este artículo; son dibujos realizados por un aficionado con la ayuda del queso y del ratón.



PROTO®

CABLES

- **PROTO Cable Centronics AMSTRAD**
  Conecta al ordenador una impresora centronics
  3.200 Ptas.
- **PROTO Cable 2ª Unidad de Disco AMSTRAD**
  Conecta al ordenador la 2ª Und. de Disco
  2.000 Ptas.
- **PROTO Cable Audio M.S.X.**
  Conecta Magnetófono a ordenad. M.S.X.
  950 Ptas.
- **PROTO Cable Audio AMSTRAD**
  Conecta Magnetófono a ord. AMSTRAD
  950 Ptas.
- **PROTO Set de Cables Prolongadores AMSTRAD 8256**
  Prolongador impresora y prolongador alimentación
  3.750 Ptas.
- **PROTO Set de Cables Prolongadores AMSTRAD 464**
  Prolongador alimentación y prolongador monitor
  1.600 Ptas.
- **PROTO Set de Cables Prolongadores AMSTRAD 664-6128**
  Dos prolongadores alimentación y prolongador monitor
  2.300 Ptas.

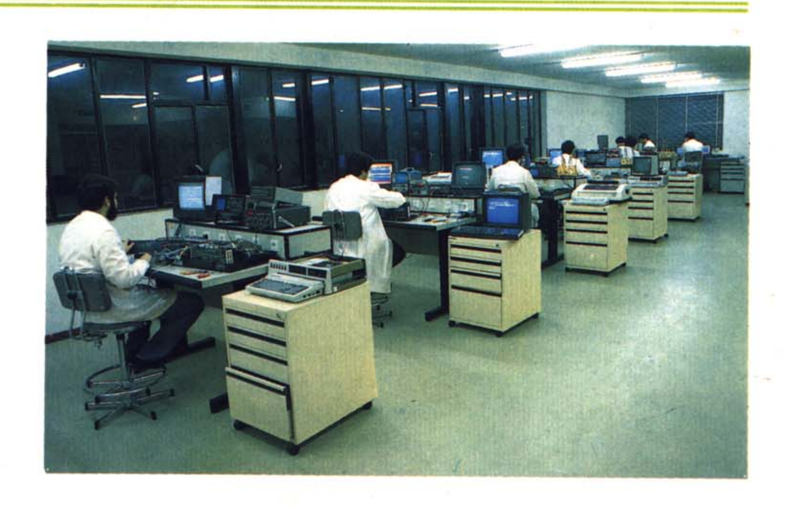
Precios Ex IVA

PROTOMEC, S.A. Avda. de la Constitución, 260 - Telf. 675 78 54 - TORREJON DE ARDOZ (Madrid)

SERVICIO TECNICO DE REPARACION DE ORDENADORES

REPARAMOS

AMSTRAD TODOS LOS MODELOS
SPECTRUM
COMMODORE
TEXAS INSTRUMENTS
SPECTRAVIDEO



PROGRAMA TRADUCTOR DE TEXTO

En uno u otro momento todos hemos sentido la tentación de «personalizar» algunos programas en lo que respecta a los textos que aparecen en pantalla.

Unas veces nos gustaría cambiar alguna palabra por otra, otras nuestra intención pudiera ser introducir rótulos a nuestro antojo, aprovechando espacios en blanco. Quizá para una mayoría sería una gran satisfacción poder disponer de una versión española «hecha por nosotros mismos».

Aunque la idea es tentadora, tiene algunas dificultades ponerla en prática cuando las zonas de texto se hallan entremezcladas con el resto del código máquina.



La primera dificultad estriba en localizar el texto dentro de la RAM, máxime si se tiene en cuenta que por lo general se halla muy fragmentado.

El siguiente problema nos lo encontramos en la propia traducción o sustitución. Ocurre algo parecido al doblaje de las películas (la frase española debe ser igual de larga y con una vocalización parecida a la del idioma original). En nuestro caso resulta totalmente imprescindible que el nuevo texto sea como máximo de la longitud del original. Si nos «salimos» podemos garantizar que el programa dejará de funcionar correctamente, en su totalidad o en alguna de sus partes.

Nos toca ahora sustituir, caracter a caracter, en la RAM el texto antiguo por el nuevo.

La experiencia nos demostrará que al rodar el programa la primera vez (y a veces alguna más...) observaremos con cierto disgusto que la cosa no nos ha quedado como deseábamos.

Quizá sea este el momento de probar el programa que os presentamos... y que se rompa la cabeza el ordenador.

Su uso es muy sencillo ya que basta con seguir paso a paso las instrucciones que van apareciendo en pantalla y por eso nos centraremos sólo sobre su estructura y filosofía.

La idea principal estriba en representar sólo aquellos caracteres ASCII que no sean «de control» (valores 1 - 31). Los caracteres de control se sustituyen en pantalla por un cuadrado relleno (código 255, ver instrucciones 9230 y 9330) para facilitar la lectura del texto. Con ayuda de RETURN y el resto de las teclas alfanuméricas podemos desplazarnos sobre el texto para cambiarlo o dejarlo como está.

Una vez finalizada la modificación del texto que deseemos, lo más recomendable es pasarla a cinta (o *diskette*) antes de tratar de ver cómo nos ha quedado rodando el programa completo. (Si el texto es largo esta maniobra debe hacerse de vez en cuando para no perder el trabajo ante un fallo de energía o un *crack* del programa). El programa incluye también

esta facilidad (ver opciones 5 y 6, instrucciones 9530 y 9570).

Es muy importante traducir o modificar sólo aquellos caracteres que sepamos con certeza que componen un texto y abstenernos cuando hay duda ya que puede tratarse de parte de instrucciones en código máquina (ojo a las letras o símbolos sueltos).

Para saltar a explorar otra parte de la RAM distinta a donde nos encontramos sólo debemos pulsar la barra espaciadora y elegir la posición de memoria a partir de la cual queremos trabajar. Si lo deseamos podemos echar un vistazo rápido a una amplia zona de memoria (opción 1) para elegir cómodamente la zona donde queremos centrar la traducción.

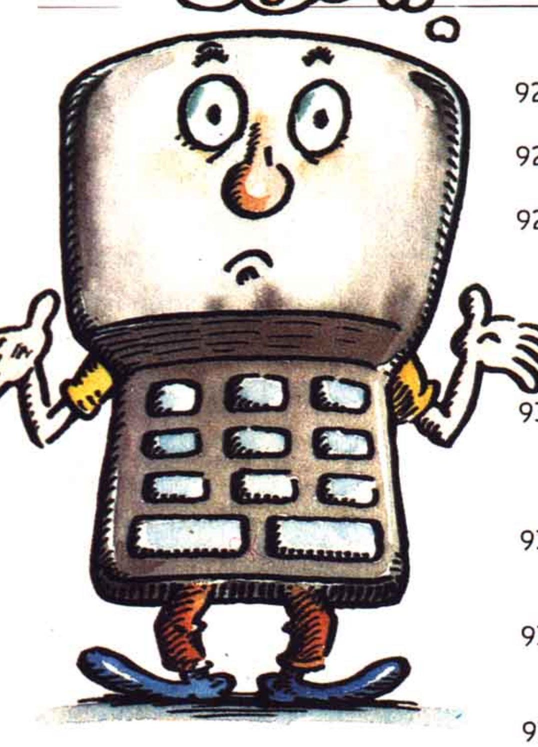
La práctica nos ha demostrado que resulta muy conveniente no perder idea del conjunto y por eso el programa permite visualizar el renglón que estamos traduciendo y además va dejando en pantalla los últimos renglones manejados.

Las opciones 3 y 4 nos sirven de ayuda para cargar en RAM la parte de C/M que deseemos traducir, tanto de *cassette* como de disco.

Por último indicaremos que el programa está preparado (la primera sentencia es la 9000) de manera que normalmente podamos situarlo (con MERGE) dentro de la parte BASIC que controla la parte C/M. Esta forma resulta cómoda para alternar la traducción con el rodaje del programa inicial para ver de forma práctica como va quedando nuestro trabajo (GO TO 9000 para activar la parte de traducción).

Felices traducciones.

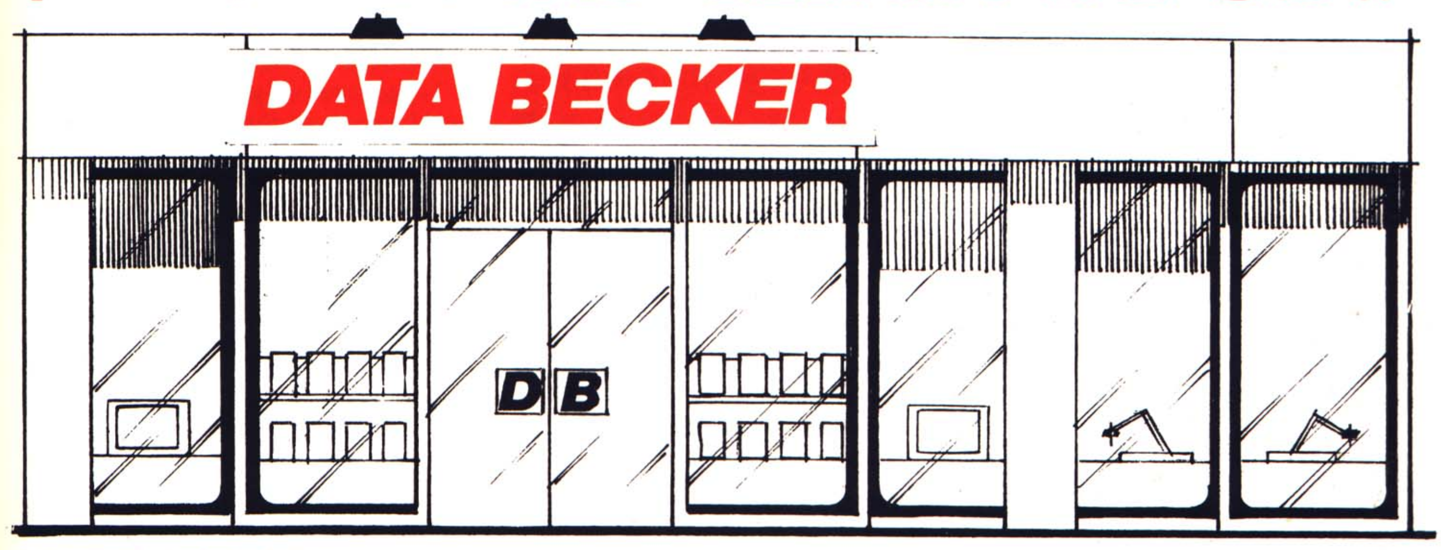
```
9000 'TRADUCTOR DE TEXTO
     EN C/M
9010 'MOLISOFT 86
9020 SCREEN 0:WIDTH 32
     :KEY OFF
9030 '
9040 'MENU
9050 '
9060 CLS:LOCATE 0,0,0:PRINT
     "PROGRAMA TRADUCTOR
     DE TEXTO":PRINT"--------
     --------- -- -----"
9070 LOCATE 0,5,0:PRINT
     "1 - VISTAZO GENERAL"
9080 LOCATE 0,7,0:PRINT
     "2 - TRADUCCION"
9090 LOCATE 0,9,0:PRINT
     "3 - CARGAR C/M DE
     CASSETTE"
9100 LOCATE 0,11,0:PRINT
     "4 - CARGAR C/M DE
     DISCO"
9110 LOCATE 0,13,0:PRINT
     "5 - ARCHIVAR C/M EN
     CASSETTE"
9120 LOCATE 0,15,0:PRINT
     "6 - ARCHIVAR C/M EN
     DISCO"
9130 OP$=INKEY$:IF OP$=""
     GOTO 9130
9140 IF ASC (OP$)<49 OR
     ASC (OP$)>54 GOTO 9130
```



```
9150 OP=VAL(OP$)
9160 ON OP GOTO 9180,9280,
     9450,9490,9530,9570
9170 '
9180 ' VISTAZO GENERAL
9190 '
9200 CLS: INPUT "POS. INICIAL
     DE MEMORIA  ";PS:IF PS<0
     OR PS>65535!  THEN GOTO
     9200
9210 PU=((PS+511)>=65535!)*
     (-65535!)+((PS+511)<
     65535!)*(- (PS+511))
9220 PRINT: PRINT "VAL. ACT.
     EN POS. ";PS;"-";PU
     :PRINT
9230 FOR I=PS TO PU :C=PEEK
     (I) :C=(C<32 ) *(-255)
     +(C>=32 )*(-C) :PRINT
     CHR$ (C);:NEXT I:PRINT:
     PRINT:PRINT"SEGUIR='
     RETURN'MENU=RESTO TECL"
9240 OP$=INKEY$:IF OP$=""
     GOTO 9240
9250 IF ASC (OP$)<>13 THEN
     GOTO  9020
9260 IF PU=65535! THEN  GOTO
     9020 ELSE PS=I:GOTO
     9210
9270 '
9280 ' TRADUCCION
9290 '
9300 CLS: INPUT "POS. INICIAL
     DE MEMORIA  ";PS:IF PS<0
     OR PS>65365! THEN GOTO
     9300
9310 PU=((PS+31)>=65535!)*
     (-65535!)+((PS+31)<
     65535!)*(- (PS+ 31))
9320 PRINT: PRINT "POSICIONES
     DE ";PS;" A ";PU
     :PRINT
9330 FOR I=PS TO PU :C=PEEK
     (I) :C=(C<32 ) *(-255)
     +(C>=32 )*(-C) :PRINT
     CHR$ (C);:NEXT I
9340 PRINT :PRINT:PRINT
     "PULSAR NUEVO VALOR O
     'RETURN'":PRINT:I=PS
9350 C=PEEK(I) :D=(C<32) *
     (-255)+(C>=32)*(-C)
     :PRINT CHR$(255);
9360 OP$=INKEY$:IF OP$=""
     GOTO 9360
9370 IF ASC (OP$)=13 THEN
     PRINT CHR$(8); CHR$(D);
     ELSE  PRINT CHR$(8); OP$
     ;:POKE I,ASC(OP$)
9380 I=I+1:IF I<=PU THEN
     GOTO 9350
9390 PRINT:PRINT: PRINT
     "SEGUIR=RETURN ** MENU=
     BARRA ESPACIADORA **
     REPETIR=RESTO TECLAS"
     :PRINT
9400 OP$=INKEY$:IF OP$=""
     GOTO 9400
9410 IF ASC (OP$)=32  THEN
     9020
9420 IF ASC (OP$)<>13 THEN
     I=PS: GOTO 9350
9430 IF PU=65535! THEN
     GOTO 9020 ELSE PS=PU+1
     :GOTO 9310
9440 '
9450 'CARGAR TEXTO C/M DESDE
     CASSETTE
9460 '
9470 CLS: INPUT "NOMBRE ";N$
     :BLOAD "CAS:"+N$:GOTO
     9020
9480 '
9490 'CARGAR TEXTO C/M DESDE
     DISCO
9500 '
9510 CLS: INPUT "NOMBRE ";N$
     :BLOAD "A:"+N$:GOTO
     9020
9520 '
9530 'ARCHIVO EN CASSETTE
9540 '
9550 CLS: INPUT "NOMBRE ";N$:
     INPUT"DIRECCION DE
     COMIENZO";DI:INPUT
     "DIRECCION FINAL";DF:
     BSAVE "CAS:"+N$,DI,DF:
     GOTO 9020
9560 '
9570 'ARCHIVO EN DISCO
9580 '
9590 CLS: INPUT "NOMBRE ";N$:
     INPUT"DIRECCION DE
     COMIENZO";DI:INPUT
     "DIRECCION FINAL";DF:
     BSAVE "A:"+N$,DI,DF:GOTO
     9020
9600 M$=INKEY$:IF M$="" THEN
     9600
9610 PRINT ASC( M$):GOTO 9600
```



¡AHORA MAS CERCA DE UD.!



VISITE NUESTRO NUEVO LOCAL EN BARCELONA:
Córcega, 299 (entre Balmes y Rambla Cataluña)

- Nuestros técnicos podrán darle amplia información sobre nuestros productos o sobre otros que consideremos punteros.
- Tendrá ocasión de probar antes de comprar.
- Todos nuestros productos están garantizados.
- En nuestro local encontrará el servicio post-venta que siempre deseó.
- ¡PONGANOS A PRUEBA! LE ESPERAMOS.

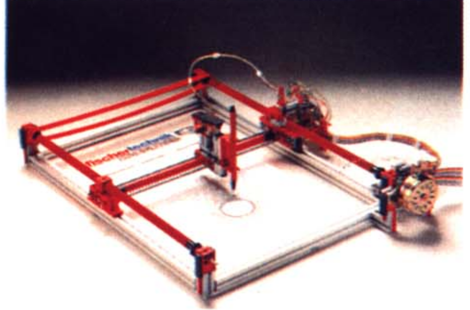
ESTAMOS EN INFORMAT 86, STAND G-9, PALACIO N.° 4



IMPORTADOR EXCLUSIVO: FERRE MORET, S. A.



30 572 Trainingsroboter



30 571 Plotter · Scanner



30 554
Baukasten computing

LOS ROBOTS DIDACTICOS QUE POR PRECIO Y CALIDAD NO TIENEN RIVAL

DATA BECKER EL N.° 1 EN INFORMATICA

FERRE MORET, S.A. - C/. TUSET, 8-10, ENTL. 2.ª TEL 218 02 93 - 08006 BARCELONA

COMO SE ALMACENAN TUS PROGRAMAS BASIC

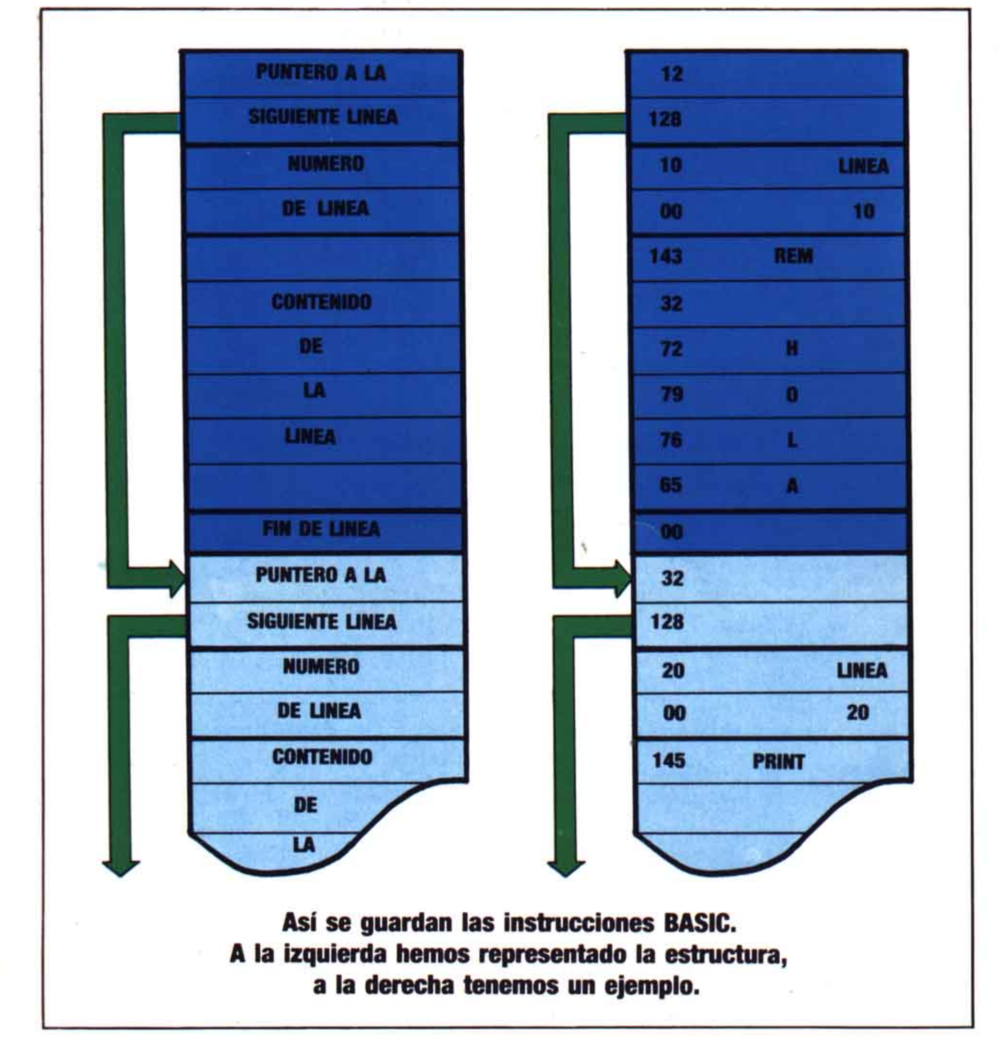
Cuando escribes un programa en lenguaje BASIC, introduces a través del teclado de tu ordenador una serie de caracteres: números, letras y símbolos especiales. ¿Cómo y dónde se almacenan estos caracteres que constituyen el programa?

Escribir un programa en lenguaje BASIC, al igual que en cualquier otro lenguaje, no es más que proporcionar al ordenador una lista de instrucciones, diciéndole qué es lo que tiene que hacer. Las instrucciones llevan un orden, es decir, le dicen al ordenador no sólo lo que tiene que hacer, sino también cuando tiene que hacerlo; algo así como: en primer lugar haz esto, en segundo lugar esto otro, a continuación...

En lenguaje BASIC, esta ordenación o secuencia de las instrucciones viene fijada por los números de línea que aparecen delante de cada instrucción. Así, una instrucción cuyo número de línea sea, por ejemplo, el 10, se ejecutará normalmente antes que otra cuyo número de línea sea el 20. Y decimos normalmente, porque puede no ocurrir así, dependiendo de las bifurcaciones o saltos del programa (GOTO, GOSUB, etc).

Todas estas instrucciones las escribe el programador desde el teclado, utilizando para ello las palabras clave del lenguaje BASIC (por ejemplo REM, PRINT o GOTO) y toda una serie de números, letras y símbolos especiales que el lenguaje pone a disposición del programador. Una vez escritas, este conjunto de instrucciones, que constituyen el programa, quedan esperando en la memoria RAM hasta que el programador decide hacer algo con ellas. Este puede iniciar la ejecución del programa (RUN) o quizá almacenarlo permanentemente en un *cassette* o *diskette* (SAVE), o puede querer un listado por impresora (LLIST) o tal vez decida borrarlo en parte (DELETE) o totalmente (NEW o apagando el ordenador).

El caso es que hasta que no llegue alguna de estas órdenes, el programa permanece tranquilamente almacenado en la memoria RAM.



Así se guardan las instrucciones BASIC. A la izquierda hemos representado la estructura, a la derecha tenemos un ejemplo.

COMO Y DONDE

Lo que nos interesa saber es cómo se almacena el programa, es decir, con qué estructura y de qué forma se organizan las instrucciones en la memoria, y también dónde se almacena el programa, esto es, entre qué posiciones o direcciones de memoria quedan esperando las instrucciones.

La segunda pregunta es la más sencilla de responder y vamos a empezar por ella. La memoria de nuestro **MSX**, al igual que todas las memorias, está organizada como una sucesión de lo que se conoce como palabras de memoria. Una palabra de memoria es una colección de bits, es decir un conjunto de estados lógicos binarios que sólo pueden tomar el valor 0 o el valor 1. En nuestro ordenador, como en la mayoría de los ordenadores domésticos, cada palabra de memoria consta de 8 bits. Hay ordenadores con palabras de más bits, pero lo normal en las máquinas pequeñas son 8 bits o lo que es lo mismo un byte.

Hay un buen montón de bytes en la memoria de nuestro ordenador; según

el modelo de **MSX** que consideremos tendremos desde unos 16384 (16K) hasta 65536 (64K) e incluso más.

Para organizar todos estos bytes y saber dónde se encuentra cada uno de ellos, todo byte lleva asignado un número de orden, que también se conoce como dirección del byte o dirección de memoria. Así, una memoria de 64K incluye desde la dirección 0 hasta la dirección 65535.

En estos bytes o direcciones de memoria es donde se almacena toda la información que maneja el ordenador en un momento dado y por supuesto, también es ahí donde se almacenan las instrucciones de nuestro programa BASIC.

¿Pero dónde? ¿En cuáles de los bytes? Podría pensarse que lo más sencillo es almacenar las instrucciones empezando desde la dirección 0 y siguiendo en direcciones crecientes. Este no es el caso. Los ordenadores **MSX** tiene ocupadas las primeras 32768 direcciones de memoria con las rutinas que forman el sistema operativo y el intérprete de BASIC. Estas rutinas van almacenadas en ROM (memoria sólo de lectura) que es un tipo de memoria permanente, que no se borra al desconectar el ordenador, pero que no se puede modificar, es decir, no podemos guardar en ella nada de nuestra información y tampoco nuestro programa BASIC.

La siguiente dirección de memoria es la 32769. Desde esta dirección nos encontramos con memoria RAM, es decir memoria en la que podemos leer y escribir, y por tanto, sobre la que podemos almacenar nuestros programas BASIC. A partir de dicha dirección y en las siguientes es donde quedan almacenadas las instrucciones de cualquier programa BASIC.

¿Cómo se almacenan las instrucciones?

Cuando nosotros escribimos una instrucción en lenguaje BASIC introducimos a través del teclado un conjunto de caracteres: letras, números y símbolos especiales. Por ejemplo supongamos que escribimos:

```
10 REM HOLA
20 PRINT "BUENOS DIAS"
```

TOKENS DEL ESTANDAR MSX

129	END	130	FOR	131	NEXT
132	DATA	133	INPUT	134	DIM
135	READ	136	LET	137	GOTO
138	RUN	139	IF	140	RESTORE
141	GOSUB	142	RETURN	143	REM
144	STOP	145	PRINT	146	CLEAR
147	LIST	148	NEW	149	ON
150	WAIT	151	DEF	152	POKE
153	CONT	154	CSAVE	155	CLOAD
156	OUT	157	LPRINT	158	LLIST
159	CLS	160	WIDTH	161	LSE
162	TRON	163	TROFF	164	SWAP
165	ERASE	166	ERROR	167	RESUME
168	DELETE	169	AUTO	170	RENUM
171	DEFSTR	172	DEFINT	173	DEFSGN
174	DEFDBL	175	LINE	176	OPEN
177	FIELD	178	GET	179	PUT
180	CLOSE	181	LOAD	182	MERGE
183	FILES	184	LSET	185	RSET
186	SAVE	187	LFILES	188	CIRCLE
189	COLOR	190	DRAW	191	PAINT
192	BEEP	193	PLAY	194	PSET
195	PRESET	196	SOUND	197	SCREEN
198	VPOKE	199	SPRITE	200	VDP
201	BASE	202	CALL	203	TIME
204	KEY	205	MAX	206	MOTOR
207	BLOAD	208	BSAVE	209	DSKO$
210	SET	211	NAME	212	KILL
213	IPL	214	COPY	215	CMD
216	LOCATE	217	TO	218	THEN
219	TAB(	220	STEP	221	USR
222	FN	223	SPC(	224	NOT
225	ERL	226	ERR	227	STRING$
228	USING	229	INSTR	230	'
231	VARPTR	232	CSRLIN	233	ATTR$
234	DSKI$	235	OFF	236	INKEY$
237	POINT	238	>	239	=
240	<	241	+	242	-
243	*	244	/	245	^
246	AND	247	OR	248	XOR
249	EQV	250	IMP	251	MOD

DE 252 A 254 NO SE REPRESENTAN TOKENS

EL RESTO DE TOKENS SE OBTIENEN CON EL PRIMER BYTE A 255 Y EN EL SEGUNDO UN VALOR ENTRE 1 Y 48

Hemos escrito un uno, un cero, un espacio,... una A y hemos pulsado RETURN, con lo que nuestra primera instrucción ha quedado almacenada en memoria. Luego hemos tecleado un dos, un cero, un espacio,... unas comillas de cierre y hemos vuelto a pulsar RETURN almacenando la segunda instrucción.

Estas dos instrucciones se almacenan con una estructura común, que va a ser la de cualquier otra instrucción que tecleemos.

Esta estructura, que hemos repre-

sentado en la figura 1, consta de cuatro partes bien diferenciadas. En primer lugar nos encontramos con un puntero que señala a la dirección de comienzo de la siguiente instrucción. Este puntero está constituido por dos bytes, los primeros de la instrucción, cuyo contenido es la dirección de memoria donde comienza la siguiente instrucción. Dicha dirección se calcula como sigue:

```
DIRECCION DE INICIO =
CONTENIDO DEL BYTE BAJO +
256*CONTENIDO DEL BYTE ALTO
```

En el caso de la primera instrucción, que hemos dicho que comienza a almacenarse a partir de la dirección 32769, el contenido de esta dirección y el de la 32770, nos indican la dirección de comienzo de la siguiente instrucción.

Para calcularla no tenemos más que teclear lo siguiente:

```
PRINT PEEK(32769)
+256*PEEK(32770)
```

Por ejemplo, si en la dirección 32769 el contenido es 11 y en 32770 el contenido es 128, la dirección de comienzo de la segunda instrucción será:

```
DIRECCION=11+256*128
=32779
```

A continuación, después de los dos bytes del puntero a la siguiente instrucción, nos encontramos con otros dos bytes con un cometido muy determinado, pero esta vez más familiar: son los que almacenan el número de línea. En nuestro ejemplo, la primera instrucción lleva el número 10, por lo que sus dos bytes de número de línea (que son el 32771 y el 32772) contendrán respectivamente los valores 10 y 0. El número de línea se calcula de nuevo como:

```
NUMERO DE LINEA=
BYTE BAJO + 256*BYTE ALTO
```

Teniendo en cuenta que cada byte puede contener un número entre 0 y 255, al aplicar la fórmula vemos que el máximo número de línea que podemos almacenar es el 65535.

Prueba a introducir una línea con un número de línea mayor, verás como el ordenador la rechaza.

La tercera parte de la estructura de cada instrucción es el contenido de la instrucción en sí. Es decir, en la primera instrucción sería REM HOLA, y en la segunda PRINT «BUENOS DIAS». Esta parte ocupará un número variable de bytes, más cuanto más larga sea la instrucción.

Por último, la última parte de la estructura de cada instrucción está constituida por un byte cuyo contenido es cero. Este byte marca el final de la línea.

El byte siguiente ya pertenece al puntero de la siguiente línea de programa.

Así de esta forma se van almacenando todas y cada una de las líneas que constituyen un programa. Unas a continuación de otras y todas con la misma estructura.

Sólo queda un detalle más. Después del último byte de la última línea, que como hemos dicho contiene un cero, hay dos bytes más, también a cero, que indican el final del programa.

Esta es la estructura de nuestro programa ejemplo y es también la de cualquier programa BASIC, ya sea corto, largo, sencillo o complicado.

Para que puedas ver con tus propios ojos la estructura de cualquier programa BASIC almacenado en la memoria de tu ordenador, hemos preparado un sencillo programa de ayuda. Es el siguiente:

```
10 REM HOLA
20 PRINT "BUENOS DIAS"
100 KEY OFF:CLS:WIDTH 40
110 D1=32769!
120 D2=PEEK(D1)+(256*
    PEEK(D1+1))
130 CLS
140 LOCATE20,0:PRINT
    "Siguiente linea"
150 PRINT D1;" ";PEEK(D1)
    :PRINT D1+1;" ";PEEK
    (D1+1):LOCATE 20,1
    :PRINT D2
160 LOCATE20,3:PRINT
    "Num. de linea"
170 PRINT D1+2;" ";PEEK
    (D1+2):PRINT D1+3;" ";
    PEEK(D1+3)
180 LOCATE 20,4:PRINT
    256*PEEK(D1+3)+PEEK
    (D1+2)
190 LOCATE 20,6:PRINT"Linea"
200 FOR N=D1+4 TO D2-1
210 PRINT N;" ";PEEK(N);
220 IF PEEK(N)<32 OR PEEK
    (N)>126 THEN PRINT TAB
    (20)"." ELSE PRINT TAB
    (20) CHR$(PEEK(N))
230 NEXT
240 LOCATE 0,23:PRINT
    "pulsa una tecla";
250 A$=INKEY$:IF A$=""
    THEN 250
260 D1=D2
270 GOTO 120
```



Las dos primeras líneas del programa, la 10 y la 20, son las que utilizabamos como ejemplo anteriormente; son sencillas, cortas y nos van a permitir llevar a cabo algún que otro experimento.

La línea 100 se encarga de las labores de limpiar la pantalla, fijar el ancho de línea, etc.

La línea 110 fija la dirección de ini-

cio de los programas BASIC (la 32769) y la asigna a la variable D1. La dirección de inicio de la siguiente línea BASIC se calcula, con la fórmula que hemos visto, en la línea 120 y queda asignada a la variable D2. De esta forma sabemos que entre los valores de D1 y D2 se encuentra almacenada la primera línea del programa.

En 140 y 150 se imprimen, cuidando el formato, los bytes que señalan al inicio de la siguiente línea. Entre 170 y 180 se imprimen los bytes que contienen el número de línea. Desde la línea 190 a la 230, el programa se encarga de imprimir el contenido de la línea en sí. A la izquierda aparece la dirección de memoria, a continuación su contenido y por último el caracter que este representa. Sólo se van a imprimir caracteres cuando el contenido de la posición de memoria está comprendido entre 32 y 126, valores que en código **ASCII** representan a los caracteres alfanuméricos normales. Cuando el contenido de la dirección sea menor que 32 o mayor que 126, en lugar del caracter correspondiente se imprime un punto.

Por último, las líneas 240 a 270 se ocupan de que el programa espere hasta que se pulse una tecla, antes de seguir adelante en su exploración de la siguiente línea BASIC.

Ahora, y con la ayuda del programa, vamos a experimentar un poco.

Si ejecutas el programa verás que el contenido de las dos primeras direcciones (32769 y 32770) es 12 y 128 respectivamente. Prueba a cambiar el valor 12 por otro valor. Para ello interrumpe el programa con CTRL/STOP y teclea:

```
POKE 32769,15
```

Ahora lista el programa y observa lo que ocurre. Al cambiar el valor 12 por el valor 15, has modificado la dirección de comienzo de la siguiente instrucción lo que ha dado lugar a la confusión que puedes ver.

Teclea ahora lo que sigue:

```
POKE 32769,12
```

Has vuelto a poner el valor correcto. Verás como al listar el programa todo vuelve a la normalidad.

Vamos a ver a continuación algo curioso que ocurre si cambiamos un número de línea. Escribe RUN y observa los contenidos de las direcciones 32771 y 32772 (10 y 0 respectivamente). Estos valores te indican que te encuentras ante la línea 10 del programa. Vamos a cambiar estos valores. Para ello interrumpe el programa (CTRL/STOP) y teclea:




```
POKE 32771,30
```

Con ello has cambiado el número de línea 10 por el número de línea 30. Para comprobarlo no tienes más que listar el programa.

Como sólo has modificado el número de línea de la primera línea y has dejado el resto de la memoria del programa intacta, observarás que sólo ha cambiado el número de la primera línea pero, y esto es lo curioso, no ha habido reordenación de lineas, por lo que te encuentras con la línea 30 antes que con la línea 20. Escribe:

```
POKE 32771,10
```

y lista de nuevo el programa. Todo estará en orden.

LOS TOKEN

Hemos dejado para el final el comentario acerca de cómo se almacena la propia instrucción, es decir, lo que queda si quitamos el puntero a la siguiente línea, el número de línea y el byte de fín de línea.

Sobre todo nos interesa lo que son palabras clave del lenguaje (REM en la instrucción 10, PRINT en la 20, etc.). Se podría pensar que la mejor forma de almacenar, por ejemplo, la palabra clave REM es utilizando tres bytes: uno para la R, otro para la E y el tercero para la M. No es así. En lugar de ello, cada palabra clave se almacena en un solo byte, utilizando un valor que identifica a la palabra clave y que se conoce como **«token»** de esa palabra.

Hemos incluido una tabla con todas las palabras clave y su **token** correspondiente. En ella podrás observar que el **token** de REM es 143, mientras que el de PRINT es 145. Esto quiere decir que en la memoria, en lugar de almacenarse los códigos de los caracteres R, E y M en tres bytes, se almacena en un solo byte el valor 143. El ordenador sabe que 143 corresponde a REM porque lleva una tabla parecida, almacenada en la memoria ROM.

Como curiosidad, ejecuta nuestro programa de experimentación y fíjate en el contenido de la dirección 32773. Es 143, es decir, el **token** del REM que aparece en la línea 10. Interrumpe el programa (CTRL/STOP) y teclea:

```
POKE 32773,145
```

Lista el programa y verás que en la instrucción 10 ha desaparecido el REM y ha sido sustituido por un PRINT. Ello se debe a que, en la memoria, hemos cambiado el **token** de REM por el de PRINT.

Prueba a cambiar el valor de 145 por otros y verás como al listar el programa aparecen diversas palabras clave BASIC sustituyendo al inicial REM.



LOS MEJORES DE INPUT MSX

PUESTO	TITULO	PORCENTAJE
1.°	Soccer	19,6 %
2.°	Knight Lore	18,8 %
3.°	H.E.R.O.	11,7 %
4.°	Yie ar kung fu	9,4 %
5.°	Profanation	8,6 %
6.°	Hyper rally	7,9 %
7.°	Boxing	7,8 %
8.°	Alien 8	7,2 %
9.°	Zakil wood	4,7 %
10.°	Hypersport I	4,3 %
		100 %

Para la confección de esta relación únicamente se han tenido en cuenta las votaciones enviadas por nuestros lectores de acuerdo con la sección «Los Mejores de Input».

Mayo de 1986



EL MUNDO MAGICO DE ORB

Cuentan las leyendas de **Orb**, isla solitaria perdida en mitad del océano, que tú, el protagonista del juego, fuiste confiado a la custodia de los monjes del **Templo de la Roca**, para que hicieran de tí un guerrero **Ninja**.
Tras muchos años de duro adiestramiento ha llegado el momento de la prueba final, el combate supremo contra los enemigos enviados por tu maestro, con el que se decidirá, con la vida o con la muerte, si eres digno de recibir el nombre de **Ninja**.
El combate no va a ser fácil. Se desarrollará en los escenarios de la tierra de **Orb** y estará dividido en tres fases: el combate sin armas, la lucha con palos y el combate con espadas *samurai*.
Entre una fase y otra del combate podrás descansar, pues cada fase viene grabada en una cara distinta de *cassette* (el juego consta de dos *cassettes*) de modo que cuando termina una fase, hay que cambiar de cara y cargar la siguiente.
En cada una de las fases vas a tener que utilizar todos los trucos que te enseñaron los monjes, si quieres salir con vida. Para ello utilizarás los distintos golpes, defensas, saltos y retiradas que podrás obtener desde tu *joystick* o desde el teclado, como prefieras. Nuestras preferencias se las lleva el teclado; el *joystick* lo hemos encontrado poco preciso pues, al haber un golpe o movimiento diferente por cada una de las ocho posiciones del *joystick*, es fácil mover éste por ejemplo hacia arriba y desviarse ligeramente en diagonal, con lo que en lugar de golpear el cuello del enemigo, daremos un salto hacia arriba. Los resultados de estas imprecisiones, que no ocurren si se utiliza el teclado, pueden ser fatales.
El grado de dificultad crecerá a medida que derrotes a más y más enemigos, hasta que al final, en la fase de lucha con espadas, te encuentres frente a frente luchando contra tu maestro.
Si consigues vencerle obtendrás el título de **Ninja**.
Los enemigos a los que te enfrentarás serán de diversos tipos, formas y colores. Habrá desde personas parecidas a tí, hasta extraños luchadores con aspecto de roca.







DATOS GENERALES

TITULO The way of the tiger

FABRICANTE Gremlin Graphics

CLASE DE PROGRAMA Artes marciales

FORMATO Cassette

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	8
INTERES	10
GRAFICOS	9
COLOR	8
SONIDO	6
TOTAL	41

Tus movimientos y los de tus enemigos puede que te resulten un tanto lentos, no hubiera estado mal un poco más de velocidad. De todos modos, esta falta de rapidez te obligará a usar de la estrategia. No sólo tendrás que decidir qué golpe vas a utilizar, sino también cuándo utilizarlo, teniendo en cuenta los retardos que se producen.
Tu estado físico en el combate aparecerá siempre reflejado en unos indicadores de aguante y fortaleza de la parte inferior de la pantalla.
Los escenarios en los que combatirás ofrecen un detalle técnico muy curioso que recibe el nombre de triple *scrolling* (algo así como el triple salto mortal o el más difícil todavía). Este efecto hace que las figuras del escenario más alejadas se desplazan más lentamente que las que están más cerca.

EL SHERIFF DE ROCA NEGRA





DATOS GENERALES

TITULO Gunfright

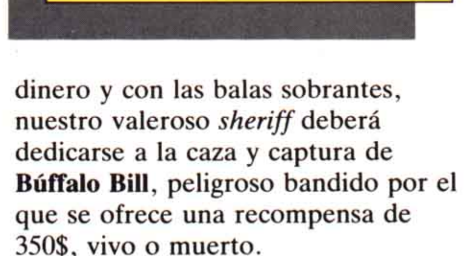
FABRICANTE Ultimate

CLASE DE PROGRAMA
Aventura en el oeste

FORMATO Cassette

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	7
INTERES	7
GRAFICOS	8
COLOR	6
SONIDO	5
TOTAL	33







Se trata de otra aventura gráfica al estilo de **Ultimate** (véase **Knight Lore** y **Alien 8**) esta vez en el escenario que proporciona **Roca Negra**, una pequeña ciudad del lejano oeste.
El protagonista de la aventura es el *sheriff* de la ciudad, un buen hombre que tiene la difícil misión de mantener la paz y el orden. La aventura comienza con una curiosa lluvia de bolsas repletas de dólares. El *sheriff* debe disparar sobre ellas para hacerse con el dinero que contienen. Sólo así podrá comprar la munición y los caballos necesarios para dar caza a los forajidos y maleantes de la ciudad.
Tras esta primera fase de recogida de dinero y con las balas sobrantes, nuestro valeroso *sheriff* deberá dedicarse a la caza y captura de **Búffalo Bill**, peligroso bandido por el que se ofrece una recompensa de 350$, vivo o muerto.
Con este objetivo, el *sheriff* tendrá que recorrer la ciudad. En las calles de la misma tropezará con pacíficos ciudadanos que surgirán desde las calles y los edificios. Los pacícificos ciudadanos resulta que no lo son tanto, ya que sólo con tocar al *sheriff* le quitan la vida. Para evitarlo puede disparar sobre ellos, pero al hacerlo verá disminuir rápidamente sus preciosos dólares.
Para recuperar el dinero perdido, el *sheriff* tendrá que recoger las bolsas de dólares que encontrará en diversos puntos de la ciudad.
Más pronto o más tarde, se producirá el encuentro con **Búffalo Bill**. Será el momento de la verdad para nuestro héroe, que no tendrá otro remedio que desafiar al forajido a un duelo a muerte. Afinando la puntería y disparando se decidirá si triunfa la ley o el crimen.
Al igual que otros programas de la misma firma, este que comentamos se desarrolla en un escenario de aspecto tridimensional, constituido por calles y edificios (en los que se puede entrar y salir) que forman un curioso e intrincado laberinto.

****

CAZANDO FANTASMAS

Sólo los cazafantasmas podrán evitar que la ciudad sea destruida por una invasión de espectros. Pero ello requerirá grandes dosis de sangre fría, habilidad, pericia y riesgo.
Lo primero es seleccionar el vehículo para desplazarnos y el equipo del que dispondremos para la detección y captura de fantasmas.
Disponemos de cuatro vehículos: el compacto, el coche fúnebre de 1968, la camioneta rural y el coche de desempeño especial. Cada uno de ellos es diferente en cuanto a la carga de equipos que puede

transportar y la velocidad que puede desarrollar. Por supuesto, su precio también varía, y deberemos hacer una elección adecuada, ya que sólo disponemos de una cantidad limitada de dinero.
A continuación escogeremos nuestro equipo. Disponemos de equipos de monitoreo y detección (el detector de energía espectral, el intensificador de imágenes y el sensor de malvavisco); de equipos de captura de fantasmas (carnada para fantasmas, trampas para fantasmas y aspiradora de fantasmas); equipo de almacenaje (sistema portátil de encierro de fantasmas por rayo láser).
Cada equipo tiene su coste en dólares, y deberemos tener cuidado en las elecciones que hacemos. De todos los elementos anteriores, el único obligatorio es la trampa de fantasmas ya que sin ella, no se puede capturar ninguno.
Con nuestro equipo, empezaremos a movernos por la ciudad, a la busca y captura de espectros. Sobre la marcha, tendremos que detener y capturar a los **Roamers**, con objeto de que no puedan llegar al templo de **Zuul** ni unirse para formar el terrible Hombre de Malvavisco.
Por cada **slimer** que capturemos aumentaremos nuestro crédito tanto más cuanto más rápidamente lo hayamos capturado. Pero deberemos ser muy precisos al capturarlos, si no, nuestros cazafantasmas serán destruidos.
El objetivo del juego es llegar al templo de **Zuul**, terrible sede de los espectros, y hacer entrar en él a dos cazafantasmas, antes de que el portero y el encargado de las llaves junten sus fuerzas en el templo de **Zuul**.
El juego es muy ameno y entretenido y con unos efectos muy bien conseguidos en gráficos, color y sonido. En resumen, un juego que aunque cuenta ya con varios años puede proporcionarnos largos ratos de diversión.





DATOS GENERALES

TITULO Ghostbusters

FABRICANTE Activision

CLASE DE PROGRAMA Juego

FORMATO Cassette

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	**9**
INTERES	**9**
GRAFICOS	**9**
COLOR	**9**
SONIDO	**8**
TOTAL	**44**





****

DE ROCA EN ROCA

La pulga iba saltando alegremente de roca en roca cuando de pronto, en un descuido, el salto se le quedó corto y, después de una larga caída, se encontró de pie en el suelo de un profundo pozo. En su caída, la pulga pudo ver cómo, a lo largo del pozo, existían una serie de rocas con entrantes, salientes y repisas.
Inmediatamente se dió cuenta de que la única forma de salir del pozo iba a ser saltando de una roca a otra, ascendiendo poco a poco hasta llegar arriba, a la boca del pozo.
En esta situación es en la que tú, jugador, te encuentras a la pobre



DATOS GENERALES

TITULO Booga-Boo

FABRICANTE Quicksilva

CLASE DE PROGRAMA Juego

FORMATO Cassette

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	9
INTERES	7
GRAFICOS	6
COLOR	7
SONIDO	4
TOTAL	33





pulga. Tu deber es el de ayudarla a salir del tenebroso pozo. Para ello vas a controlar sus saltos, decidiendo su dirección (derecha o izquierda) y su potencia (desde un saltito corto hasta un verdadero vuelo de varios metros). Los saltos deben ir adecuadamente dirigidos para colocar a la pulga en lugar seguro, en alguna de las repisas de roca, teniendo buen cuidado de que la pobrecita no resbale. Si lo hace, lo más probable es que se deslice de nuevo hasta el mismísimo fondo del pozo y haya que volver a empezar, o tal vez algo peor; puede caer en las fauces de una de las plantas comedoras de pulgas que hay en el pozo.

Además de saltar, habrá que vigilar el entorno y estar preparados para el ataque del dragón volante, animal insectívoro al que le encanta comer pulgas.

Este es el resumen del argumento de **Booga-Boo** (La pulga), un estupendo juego para tu **MSX**. Es la misma versión del programa *Made in Spain* que apareció hace ya algunos años y que llego a convertirse en número uno en Inglaterra. La versión es estupenda (la unica pega es un *scrolling* un tanto brusco) y aunque el juego no es nuevo, podemos afirmar que es una pieza que no debe faltar en ninguna buena biblioteca de programas.

★★★★★★★★★★★★★★★★★★★★★★★★★

PISA EL ACELERADOR

DATOS GENERALES

TITULO Road Fighter

FABRICANTE Konami

CLASE DE PROGRAMA Carrera de coches

FORMATO Cartucho ROM

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	6
INTERES	7
GRAFICOS	6
COLOR	7
SONIDO	7
TOTAL	33

Con **Road Fighter** tenemos la oportunidad de convertir nuestro ordenador **MSX** en un vehículo de fórmula 1.

Tendremos que pilotar a través de una serie de etapas en distintos circuitos, sorteando a todos los vehículos rivales, hasta llegar victoriosos e indemnes a la meta.

Las etapas a cubrir son seis. Cada una de ellas se desarrolla en una carretera que discurre por un paisaje diferente. Hay que contar con acantilados, despeñaderos, tramos helados, puentes, bosques... Cada tramo tiene sus particularidades y, por supuesto, sus peligros.

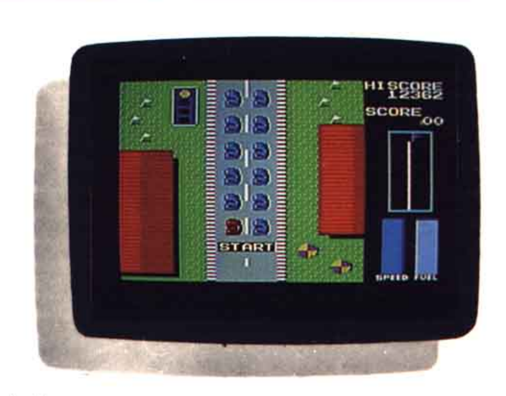
Nuestro objetivo es adelantar a todos los pilotos rivales que van en automóviles azules, y conducir nuestro automovil rojo hasta la meta. Además, habrá que tomar precauciones frente al resto de vehículos que circulan por la carretera. Algunos de ellos, por ejemplo los coches GT, los rodadores violentos y los coches de 18 ruedas, intentarán por todos los medios que nos salgamos de la carretera.

En cada tramo, aparecerán de improviso y en medio de la carretera lo que podemos definir como corazones de la suerte. Si conseguimos hacernos con ellos, obtendremos puntos y combustible extra, este último imprescindible para llegar a la meta.

Dentro de su sencillez, **Road Fighter** resulta bastante divertido. Tecnicamente cuenta con efectos bastante logrados, como el sonido del motor del coche o los efectos de aumento de velocidad. Los gráficos, en los que se representa una vista aérea del coche y del tramo de circuito por el que circula, son un tanto flojos.







★★★★★★★★★★★★★★★★★★★★★★★★

DOS BAJO PAR

No cabe duda que con la práctica asidua de este juego acabaremos convertidos en expertos en golf. Y ello porque el programa tiene en cuenta todos los factores que pueden influir en un torneo de golf auténtico; el tipo de palo a utilizar en cada momento, la fuerza del golpe, los obstáculos (arena, *bunkers*), la velocidad del viento o la pendiente del terreno en las inmediaciones del hoyo.
Todas estos factores tendremos que tenerlos en cuenta, tal y como lo haría cualquier jugador en una situación real, a la hora de elegir entre las opciones de cada golpe (palo a utilizar, dirección del golpe y fuerza del mismo).
Uno de los aspectos más curiosos y mejor realizados del programa, lo constituye la forma de presentación en pantalla de los distintos escenarios. A la derecha nos encontramos con una panorámica general del hoyo en que nos encontramos y a su lado, más grande, con una panorámica local de la zona en la que se encuentra la pelota. Por otro lado, en la parte superior de la pantalla aparece toda la información relativa al estado actual de nuestro juego; hoyo en el que nos encontramos, golpes efectuados, velocidad y dirección del viento, potencia del disparo, etc.
Podemos elegir entre jugar el partido a número de golpes o por hoyos y también si queremos jugar contra el ordenador o contra otro jugador.
El recorrido consta de 9 hoyos que deben completarse, idealmente, en 36 golpes (al principio el número de golpes se dispara y es fácil hacer 80 o más). Hay 13 palos distintos, entre los que tendremos que elegir en función de la distancia a la que queramos lanzar la pelota. Habrá que tener cuidado al realizar la elección ya que si nos equivocamos, corremos el riesgo de enviar la pelota fuera de los límites del campo (lo que significará una penalización de dos golpes) o hacerla caer en una zona difícil (arena, bosque).
Aspecto destacable del juego es el apartado de gráficos, que incluye unos efectos de perspectiva francamente interesantes.
No cabe duda de que podremos pasar tardes bastante divertidas organizando, en casa, auténticos campeonatos de golf.





DATOS GENERALES

TITULO Golf

FABRICANTE Konami

CLASE DE PROGRAMA Torneo de golf

FORMATO Cartucho ROM

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	9
INTERES	8
GRAFICOS	8
COLOR	8
SONIDO	7
TOTAL	40

CUIDADO CON LA RADIACION

Pastfinders es el nombre de un selecto grupo de personas que, en el año 8878 de nuestra era, todavía viven la aventura de las grandes exploraciones.
Nosotros podemos pertenecer a ese grupo y dedicarnos a explorar con nuestro vehículo **Leeper** una tierra desconocida y deshabitada por el exceso de radiación.
Para ello dispondremos, al principio del juego, de un mapa, en el que marcaremos la dirección del sector que queremos explorar. En dicho mapa veremos las zonas contaminadas con alta radiación, con radiación media y con baja radiación, marcadas en rojo, amarillo y verde respectivamente.
Una vez hemos elegido el sector que queremos explorar, escogeremos el equipo de defensa y ayuda que queremos llevar con nosotros. La lista del equipo se compone de:
METAL PESADO: Disminuye la velocidad en la que el **Leeper** absorbe la radiación.
EQUIPOS ANTIRRAYOS: Para proteger al **Leeper** del fuego enemigo.



DATOS GENERALES

TITULO Pastfinder

FABRICANTE Activision

CLASE DE PROGRAMA Juego

FORMATO Cassette

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	6
INTERES	7
GRAFICOS	7
COLOR	7
SONIDO	6
TOTAL	33



NEUTRALIZADOR DE RADIACIONES: Permite eliminar la radiación absorbida hasta el momento.
DISTORSIONADOR: Equipo de camuflaje, para evitar ser visto por el enemigo.
Debemos ser cuidadosos cuando escojamos el equipo ya que, si el **Leeper** es destruido, todo el equipo que llevemos con nosotros se perderá.
Una vez seleccionados el sector a explorar y el equipo, comenzará la aventura. Nuestra nave se moverá por el sector (siempre hacia delante y hacia los lados, nunca podremos retroceder), y deberemos buscar y recoger unos objetos de forma circular, depositándolos en las bases y estaciones.
Pero no todo iba a ser tan sencillo. Para dificultar nuestra búsqueda habrá obstáculos, naves agresivas que nos disparan. Además, continuamente, estaremos absorbiendo radiación, pero podremos paliar este efecto destruyendo unos pequeños cristales flotantes distribuidos a lo largo de todo el sector.
Por cada artefacto que depositemos en una base obtendremos un segundo libre de radiación, y si lo depositamos en una estación, un segundo de invulnerabilidad frente a las naves enemigas y obstáculos.
Podremos además buscar y recoger **Leepers** extra y equipos de repuesto.
Cuando terminemos de explorar un sector, aparecerá de nuevo el mapa para que volvamos a marcar otro sector de exploración.
El efecto de movimiento y colores de las naves y de los obstáculos móviles está bien logrado, y el juego es entretenido, aunque su idea central no sea excesivamente original.

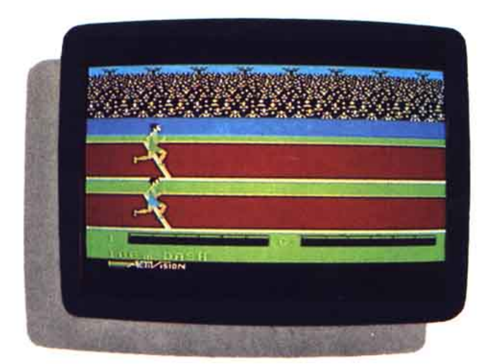
****

EN SUS MARCAS...

Si nos gustan las competiciones deportivas, ahora tendremos la oportunidad de convertirnos en campeones de atletismo, practicando las 10 pruebas del decathlon que nos ofrece este juego.
Tenemos además la posibilidad de competir nosotros solos con el ordenador, o bien de medir nuestras fuerzas con otro competidor.
Las pruebas en las que tendremos que competir son carreras (100 m, 400 m, 1500 m, 110 m vallas), saltos (longitud, altura y pértiga) y lanzamientos (de peso, de disco y de

jabalina). Cuanto mejores marcas obtengamos, más puntos se nos darán. Además, si obtenemos una buena marca, seremos obsequiados con una marcha triunfal y los aplausos del público.
La potencia y velocidad necesarias se consiguen moviendo velozmente el joystick a derecha y a izquierda.
La sensación de movimiento de los atletas está muy bien conseguida, si bien la calidad del dibujo y los efectos de sonido podrían haberse mejorado un poco.

En resumen, con **Decathlon** tenemos un divertido juego de competición, para practicar solo o en compañía, y con el que podremos practicar todo tipo de deportes... sin movernos de casa.







DATOS GENERALES

TITULO The Activision Decathlon

FABRICANTE Activisión

CLASE DE PROGRAMA Juego

FORMATO Cassette

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	8
INTERES	7
GRAFICOS	7
COLOR	7
SONIDO	6
TOTAL	35

****

A PUÑETAZO LIMPIO

Cuando te encuentres cargado de tensión y para descargarte estés pensando en pegar a alguien, olvídalo, es mejor que cojas el cartucho con este programa, **Boxing**, y dejes tranquilas a las personas que te rodean.
Si has tomado esta sabia decisión, nada más conectar el cartucho te encontrarás de pié en un cuadrilátero, frente a frente con tu oponente, mientras contemplas cómo el árbitro señala el comienzo del primer asalto. Es el momento de pegar al adversario. Utiliza los diversos golpes que puedes dar y que incluyen desde directos a la mandíbula de tu contrincante, hasta golpes en el estómago, pasando por formidables ganchos de izquierda.
Claro que, lo más normal es que tu adversario haga lo mismo contigo, así que también vas a tener que utilizar tus defensas cubriéndote la cara o el estómago, dependiendo de con qué parte de tu cuerpo se haya encariñado el bruto que tienes enfrente.
Puedes escoger entre pegarte con el ordenador o hacerlo contra otro ser humano siempre, claro está, a través de la opción para dos jugadores.
El combate terminará cuando el árbitro decrete **KO** (al no levantarse alguno de los púgiles tras la cuenta de diez), **TKO** o **KO** técnico (tras caer al suelo tres veces consecutivas) o bién al terminar el tercer asalto, en



DATOS GENERALES

TITULO Boxing

FABRICANTE Konami

CLASE DE PROGRAMA Combate de boxeo

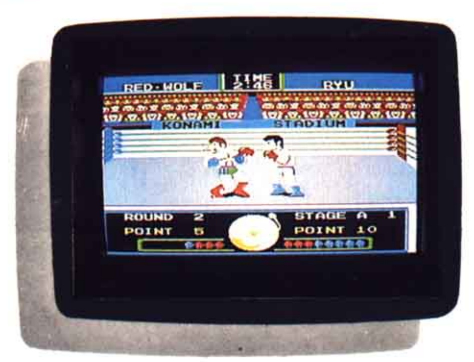
FORMATO Cartucho ROM

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	8
INTERES	7
GRAFICOS	8
COLOR	8
SONIDO	6
TOTAL	37



cuyo caso el árbitro decidirá quién ha ganado a los puntos.



El juego se desarrolla a través de una serie de diferentes combates, cada uno más difícil que el anterior, a los que irá accediendo el jugador a medida que deje fuera de combate a los contrarios.
Lo mejor del juego está en: la variedad de golpes a utilizar, las simpáticas caras del boxeador que recibe un directo en la mandíbula, el excelente sonido de los puñetazos y, en fin, los estupendos gráficos y la mejor animación.
Seguro que después de unos cuantos asaltos te encuentras totalmente relajado.

★★★★★★★★★★★★★★★★★★★★★★★★★

JOE EL PANADERO

Konami nos transporta a un escenario simpático. Una panadería industrial en la que hay que fabricar todo tipo de piezas de pan, bollos, etc. La fabricación está organizada mediante una cadena de máquinas que trabajan en serie.
La primera de las máquinas da forma a las piezas de pan, a partir de la masa de harina y agua. La segunda se ocupa de dar un primer horneado. Por último, en la tercera de las máquinas, se produce el horneado final, que dará un estupendo color dorado a todas las piezas. Después de la última máquina está esperando el camión de reparto, encargado de llevar el pan y los bollos hasta el público. Este público está constituido por cuatro niños zampabollos, con un hambre feroz.
¿Dónde entra aquí el jugador? Bueno, hay que seguir contando la historia. Resulta que en la cadena de fabricación se han colado unos traviesos mapaches, que se dedican a parar las máquinas y a zamparse los bollos antes de que lleguen al camión de reparto. El jugador asume el papel de **Joe**, el dueño de la panadería, con la misión de mantener las máquinas en funcionamiento y expulsar a los mapaches de la panadería.
Joe cuenta con un arma prácticamente infalible, sus poderes hipnóticos, que dormirán durante un cierto tiempo a los mapaches.

La jornada comienza en la panadería a las 9:00 AM. Desde esa hora y hasta las 5:00 PM, **Joe** debe intentar producir todo el pan que pueda. La cuestión de la producción es tan seria, que si no hay al menos cuatro piezas al cabo de la jornada, **Joe** perderá una vida. Por el contrario,



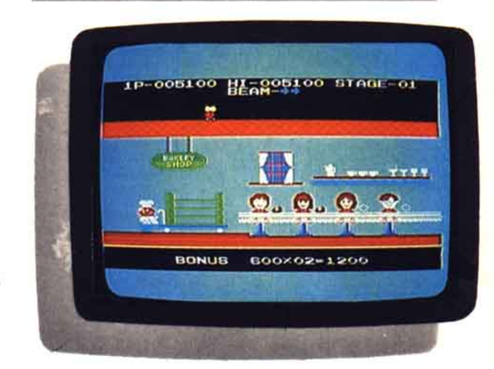
por cada pieza en exceso que produzca, obtendrá una buena cantidad de puntos extra, que le llevarán a conseguir nuevas vidas.
Jugar con **Joe** y convertirse en panadero es una verdadera delicia.
La panadería es un alarde gráfico y musical (una tonadilla te acompañará durante la jornada de trabajo). Al principio es fácil controlar a los mapaches, pero a medida que pasan los días el problema de los animalitos se vuelve más y más difícil, hay más mapaches y son más rápidos apagando las máquinas y comiéndose la producción. Inevitablemente llegará el momento de perder todas las vidas y terminará el juego.
Seguro que después de jugar un par de veces te entrará el gusanillo de conseguir mayor puntuación y te verás convertido en un adicto a la fabricación de pan.

DATOS GENERALES

TITULO Comic Bakery
FABRICANTE Konami
CLASE DE PROGRAMA Juego
FORMATO Cartucho ROM

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	8
INTERES	8
GRAFICOS	7
COLOR	9
SONIDO	8
TOTAL	40



CONCURSO INPUT/ANAYA MULTIMEDIA

¡DESCUBRE AL ASESINO!

BASES:

Durante tres números sucesivos irán apareciendo en **INPUT** las tres partes del relato «El caso del anciano asesinado» junto con los mensajes cifrados que constituyen las respuestas parciales a la solución del crimen. Cada mes, quienes logréis descifrarlos participaréis en un sorteo consistente en 10 lotes de libros por un valor de 15.000 ptas., a elegir del fondo editorial de **Anaya Multimedia**, y una suscripción gratuita por un año a **INPUT**. Habrá tres sorteos, de tal forma que no será necesario haber descifrado el enigma del mes anterior para optar al premio.

Un fabuloso premio será la guinda final de este pastel. El descubridor de las motivaciones del crimen visitará al próximo **PCW Show** a celebrar en Londres. En caso de haber más de un acertante, recurriremos a la inevitable fórmula de sorteo.

La solución del primer mensaje cifrado deberéis enviarla, junto con vuestros datos personales, a la Redacción de **INPUT** antes del 15 de junio.

Las decisiones del jurado serán inapelables, dándose en las páginas de **INPUT** cumplida cuenta de la marcha del concurso.



Nota: El libro de reciente aparición «Códigos y claves secretas», de ***Anaya,*** *contiene todos los programas útiles para esta labor. Aunque si has trabajado para los servicios de inteligencia de alguna potencia no te será de gran utilidad.*



EL CASO DEL ANCIANO ASESINADO

La muerte es una forma de desorden; cae sobre la vida como el hacha sobre el tronco y quiebra siempre la sintaxis que después es preciso recomponer con dolor, a veces con recuerdos y, casi siempre, con notables dosis de olvido.

En otras ocasiones, la muerte es una estupidez o el resultado de varias estupideces hábilmente engarzadas por el azar o por la ambición. Investigando este tipo de muerte me gano la vida.

Aquella tarde, la primavera se había desplomado sobre las calles con una bola de fuego en sus entrañas. Me llamaron por teléfono y tuve que atravesar la ciudad en un momento especialmente difícil. Los conductores, todavía con sus ropas de invierno, se asaban de calor dentro de sus coches, lo que les ponía particularmente agresivos. Los semáforos, por el contrario, parecían haber sido tratados con tranquilizantes y pasaban del rojo al ámbar y del ámbar al verde con una parsimonia que desesperaba todavía más a los sufridos conductores.

Al fin llegué a una zona residencial y subí a un piso de cinco estrellas que acababan de abandonar los bomberos. Mi ayudante ya estaba allí, intentando preservar las escasas huellas que se habían resistido a la violencia de éstos. El incendio había sido de escasa magnitud, pero había bastado para asfixiar a un pobre viejo que permanecía atado a una de las sillas del salón. Una caja fuerte, hasta entonces oculta por un cuadro, aparecía abierta y desvalijada. Los bordes de la puerta de acero presentaban señales de violencia. Anoté en uno de mis códigos cifrados:

```
QE GZM HUAXQZOUM RUZSUPM BGQE EQ FDMFM P
Q GZ FUBA PQ OMVM UZMOOQEUNXQ M OGMXCGUQ
D FUBA PQ BMXMZOM
```

Al parecer, el incendio no había sido intencionado. Según los expertos, el origen fue una colilla que, tras resbalar del cenicero, había ido a caer al pie de una cortina originando un incendio de pequeñas dimensiones, pero con un humo lo suficientemente espeso para asfixiar a un anciano y hacer llegar a los bomberos.

Juan José Millas

MSX

DYNADATA

NO LO DUDE

PRECIO INSUPERABLE

CONJUNTO: ORDENADOR MONITOR Y CASSETTE DYNADATA MSX **62.500**

TARJETA PROGRAMA. EL NUEVO FORMATO DEL FUTURO.

MONITOR 12" FOSFORO VERDE

ADAPTADOR TARJETA

DATA CASSETTE

NUEVO

DYNADATA MSX CON TECLADO EN



* Letras y signos iguales al teclado del PC de IBM.

Ordenador DYNADATA MSX con teclado en español	46.900
DYNADATA MSX con cassette y monitor de color	95.500
DYNADATA MSX y unidad de diskette de 5 ¼" de 360 Kbytes con monitor de fósforo verde	108.900
con monitor de color	141.000
TARJETA/PROGRAMA del juego LE MANS con adaptador, el cual sirve para cualquier tarjeta que usted adquiera. Precio especial con la compra del DYNADATA MSX	4.900

Con el DYNADATA MSX usted podrá:

- Divertirse con la amplia gama de juegos MSX.
- Aprender Informática y Basic con el curso autodidáctico y audiovisual.
- Llevar gestiones administrativas con los programas de proceso de textos, base de datos, contabilidad, stock, recibos, etc.
- Ayudar a sus hijos en sus estudios de 3.º a 8.º de EGB con los programas de Matemáticas, Lenguaje y Ciencias Naturales.
- Aprender idiomas tan necesarios como el inglés.
- Programar con los lenguajes: LOGO, PASCAL, FORTRAN y COBOL.

CURSO DE INFORMATICA Y BASIC

- Autodidáctico
- Audiovisual
- 12 cassettes
- 24 lecciones
- Evaluaciones periódicas
- Diploma Fin de Curso



Especificaciones DYNADATA MSX: Procesador Z80A, 64 Kbyte RAM, 16 Kbyte VRAM, 32 Kbyte ROM, 24 líneas x 40 columnas, 256 x 192 pixels, 16 colores, MSX-BASIC, MSX-DOS.

DYNADATA

Y:

SONY, PHILIPS, CANON, SANYO, JVC, TOSHIBA, SPECTRAVIDEO, PIONEER, YAMAHA, MITSUBISHI, GOLDSTAR, SAMSUNG, HITACHI, MATSUSHITA, CASIO

Se han decidido por MSX. Esto le permite compartir los programas y periféricos con todas estas reconocidas marcas.

Por todo, NO LO DUDE. Decídase por



Solicite información: Sor Angela de la Cruz, 24 - 28020 Madrid. Tels. (91) 279 21 85 - 279 28 01 - 270 01 93. Telex 44619 DYNA
Delegación Barcelona: Aribau, 61, entlo - 08011 Barcelona. Tels. (93) 254 73 04 - 254 73 03